

MIIA0106
Python and C Programming Language

อาจารย์ สุทิศ องอาจ

ภาคการศึกษา 2 ปีการศึกษา 2567

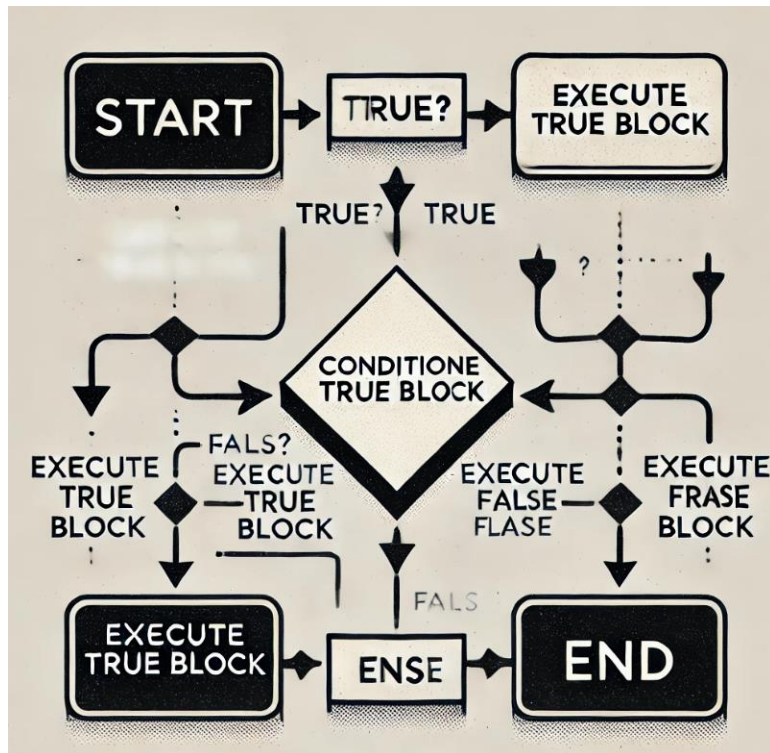
MIIA0106	Python and C Programming Language		SUN-เช้า-23/03/2568
A	MON/2	D604	[CPA/1,CPAI/1,EMAT/1]#ALL#
	MON/3	MII203	
	MON/3	MII202 A	
	MON/3	MII202 B	
B	SAT/2	MII207 B	[CPA+/1,EMAT+/1,EMA/2]#ALL#
	SUN/1	MII202 A	
	SUN/1	MII202 B	
EP	- /0	-	

Update V1 2024-11-16

Update V2 2024-11-20

2. คาบที่ 2 และ คาบที่ 3: คำสั่งตัดสินใจ, วนลูป และฟังก์ชัน (6 ชม.)

2.1. บรรยาย (3 ชม.)



หมายเหตุ : ออกแบบรูปโดย ChatGPT 4o

2.1.1. พื้นฐาน การแสดงผลใน C++ และ Python

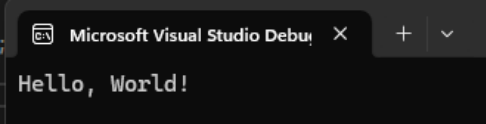
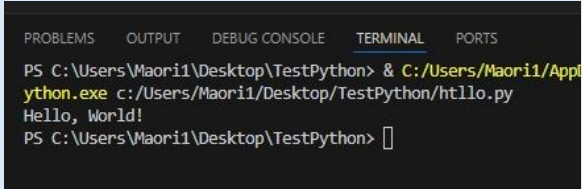
เปรียบเทียบ C++ กับ Python		
คุณสมบัติ	C++	Python
รับข้อมูล (Input)	cin >>	input()
แสดงผล (Output)	cout <<	print()
การจัดฟอร์แมตข้อความ	ต้องใช้ << และระวังช่องว่าง	ใช้ f-string ง่ายและยืดหยุ่น
การแปลงข้อมูล (Casting)	ทำงานโดยตรง (int, float, ฯลฯ)	ต้องใช้ int(), float() ฯลฯ

ความเข้าใจเบื้องต้น

C++: เป็นภาษาโปรแกรมเชิงวัตถุ (Object-Oriented Programming) ที่ให้การควบคุมระดับต่ำ (low-level) ซึ่งหมายถึงการเข้าถึงฮาร์ดแวร์ได้โดยตรง ทำให้โปรแกรมที่เขียนด้วย C++ มีประสิทธิภาพสูง แต่ก็มีความซับซ้อนในการเขียนมากกว่า

Python: เป็นภาษาโปรแกรมระดับสูง (high-level) ที่มีไวยากรณ์ที่อ่านง่ายและเข้าใจได้ ทำให้เขียนโปรแกรมได้รวดเร็วและสะดวกสบาย แต่ประสิทธิภาพอาจสู้ C++ ไม่ได้

ตัวอย่างการแสดงผลข้อความบนหน้าจอ

<p>C++ Code:</p> <pre>#include <iostream> using namespace std; int main() { cout << "Hello, World!" << endl; return 0; }</pre> 	<p>Python</p> <pre>print("Hello, World!")</pre> 
---	--

คำอธิบาย:

<ul style="list-style-type: none">• <code>#include <iostream></code>: นำ header file ที่มีฟังก์ชันสำหรับการรับเข้าและส่งออกข้อมูลมาใช้• <code>cout</code>: เป็น object ที่ใช้สำหรับแสดงผลข้อความบนหน้าจอ• <code>endl</code>: เป็น manipulator ที่ใช้สำหรับย่อบรรทัด (newline)	<p>รายละเอียดของโค้ด</p> <ol style="list-style-type: none">1. <code>print</code><ul style="list-style-type: none">○ เป็นฟังก์ชันในตัว (built-in function) ของ Python○ ใช้สำหรับแสดงผลข้อมูลหรือข้อความบนหน้าจอ2. <code>"Hello, World!"</code><ul style="list-style-type: none">○ ข้อความ (string) ที่อยู่ในเครื่องหมายคำพุดคู่ (") ซึ่งจะถูกส่งไปยังฟังก์ชัน <code>print</code> เพื่อแสดงผล
---	--

	<ul style="list-style-type: none"> o "Hello, World!" เป็นข้อความตัวอย่างที่นิยมใช้ในโปรแกรมแรกสำหรับการเริ่มเรียนภาษาโปรแกรม <p>3. การทำงาน</p> <ul style="list-style-type: none"> o คำสั่ง <code>print("Hello, World!")</code> จะส่งข้อความ "Hello, World!" ไปยังหน้าจอคอนโซล และแสดงผลออกมา
--	---

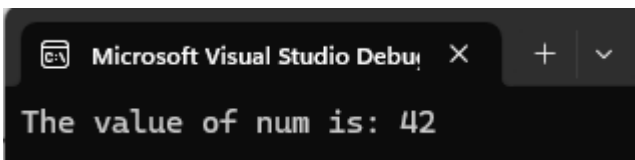
ตัวอย่างการแสดงผลตัวเลข

C++ Code:

```
#include <iostream>

using namespace std;

int main() {
    int num = 42;
    cout << "The value of num is: " << num << endl;
    return 0;
}
```



คำอธิบาย:

- 1) `#include <iostream>`
เป็นคำสั่งสำหรับการนำเข้าไฟล์ส่วนหัว (header file) `iostream` ซึ่งให้โปรแกรมสามารถใช้งานคำสั่งสำหรับการรับ (input) และส่งออก (output) ข้อมูล เช่น `cin` และ `cout`
- 2) `using namespace std;`
คำสั่งนี้ทำให้โปรแกรมสามารถใช้ฟังก์ชันใน namespace `std` ได้โดยไม่ต้องเขียน `std::` นำหน้า เช่น `cout` และ `endl`
- 3) `int main()`

เป็นฟังก์ชันหลักของโปรแกรมที่ทำหน้าที่เป็นจุดเริ่มต้นเมื่อโปรแกรมเริ่มทำงาน

int ระบุว่าโปรแกรมจะส่งค่าคืนให้ระบบปฏิบัติการในรูปแบบตัวเลขจำนวนเต็ม

4) `int num = 42;`

ประกาศตัวแปรชื่อ `num` ชนิด `int` (จำนวนเต็ม) และกำหนดค่าเริ่มต้นเป็น 42

5) `cout << "The value of num is: " << num << endl;`

ใช้ `cout` (console output) เพื่อพิมพ์ข้อความ "The value of num is: " และค่าของตัวแปร `num` ซึ่งคือ 42

ใช้ `<<` เพื่อเชื่อมข้อความและค่าที่ต้องการพิมพ์

`endl` ใช้สำหรับขึ้นบรรทัดใหม่ (new line)

6) `return 0;`

ฟังก์ชัน `main` ส่งค่ากลับ 0 เพื่อบอกระบบปฏิบัติการว่าโปรแกรมทำงานสำเร็จเรียบร้อย

Python

```
num = 42
print("The value of num is:", num)
```

```
The value of num is: 42
PS C:\Users\Maori1\Desktop\TestPython> 
```

รายละเอียดของโค้ด

1. `num = 42`

- สร้างตัวแปรชื่อ `num` และกำหนดค่าให้เป็น 42
- ตัวแปรใน Python ไม่ต้องระบุชนิดข้อมูล (data type) ชัดเจน เพราะ Python จะกำหนดชนิดข้อมูลโดยอัตโนมัติจากค่าที่กำหนดให้
- ในกรณีนี้ `num` เป็นชนิด `int` (จำนวนเต็ม)

2. `print("The value of num is:", num)`

- ใช้ฟังก์ชัน `print` (เป็นฟังก์ชันในตัวของ Python) เพื่อแสดงข้อความและค่าของตัวแปร `num` บนหน้าจอ

- ข้อความ "The value of num is:" เป็นสตริง (string) ที่จะแสดงผลตามที่เขียน
- เครื่องหมายจุลภาค (,) ใช้เพื่อแยกค่าที่ต้องการแสดงออกไปยังฟังก์ชัน print:
 - "The value of num is:" คือข้อความที่ต้องการแสดง
 - num คือค่าของตัวแปรที่จะแสดงผลตามค่าที่กำหนดไว้ (ในที่นี้คือ 42)

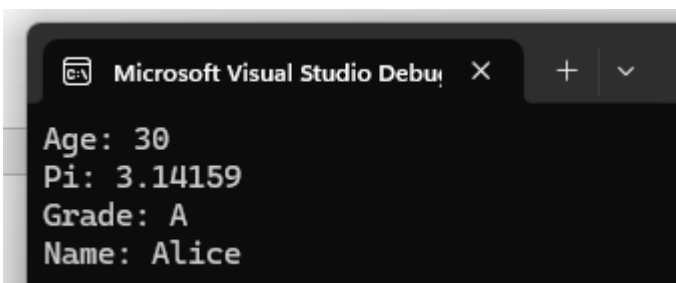
ตัวอย่างการแสดงผลข้อมูลชนิดต่างๆ

```
#include <iostream>

using namespace std;

int main() {
    int age = 30;
    double pi = 3.14159;
    char grade = 'A';
    string name = "Alice";

    cout << "Age: " << age << endl;
    cout << "Pi: " << pi << endl;
    cout << "Grade: " << grade << endl;
    cout << "Name: " << name << endl;
    return 0;
}
```



รายละเอียดของโปรแกรม

1) #include <iostream>

นำเข้าไฟล์ส่วนหัว iostream เพื่อใช้งานการรับข้อมูล (input) และแสดงผลข้อมูล (output) เช่น cin และ cout

2) using namespace std;

ทำให้สามารถใช้ฟังก์ชันใน namespace std ได้โดยตรง เช่น cout, endl และ string โดยไม่ต้องเติม std:: ข้างหน้า

3) ตัวแปร (Variables)

int age = 30;

ประกาศตัวแปรชนิด int สำหรับเก็บจำนวนเต็มและกำหนดค่าเริ่มต้นเป็น 30

double pi = 3.14159;

ประกาศตัวแปรชนิด double สำหรับเก็บเลขทศนิยมและกำหนดค่าเริ่มต้นเป็น 3.14159

char grade = 'A';

ประกาศตัวแปรชนิด char สำหรับเก็บตัวอักษรตัวเดียว (A) ใช้เครื่องหมายเดี่ยว (') ล้อมรอบ

```
string name = "Alice";
```

ประกาศตัวแปรชนิด string (ข้อความ) และกำหนดค่าเริ่มต้นเป็น "Alice" ใช้เครื่องหมายคู่ (") ล้อมรอบ

4) การแสดงผล

```
cout << "Age: " << age << endl;
```

แสดงข้อความ "Age: " และค่าของตัวแปร age (30) บนหน้าจอ

endl ทำให้ขึ้นบรรทัดใหม่หลังพิมพ์ข้อมูล

```
cout << "Pi: " << pi << endl;
```

แสดงข้อความ "Pi: " และค่าของตัวแปร pi (3.14159)

```
cout << "Grade: " << grade << endl;
```

แสดงข้อความ "Grade: " และค่าของตัวแปร grade (A)

```
cout << "Name: " << name << endl;
```

แสดงข้อความ "Name: " และค่าของตัวแปร name ("Alice")

5) return 0;

ฟังก์ชัน main คืนค่า 0 ให้ระบบปฏิบัติการ เพื่อบอกว่าการทำงานของโปรแกรมเสร็จสมบูรณ์และไม่มีข้อผิดพลาด

Python

```
age = 30
pi = 3.14159
grade = 'A'
name = "Alice"

print("Age:", age)
print("Pi:", pi)
print("Grade:", grade)
print("Name:", name)

print(f"Age: {age}")
print(f"Pi: {pi}")
print(f"Grade: {grade}")
print(f"Name: {name}")
```

```
PS C:\Users\Maori1\Desktop\TestPython> & C:/Use
python.exe c:/Users/Maori1/Desktop/TestPython/ht
Age: 30
Pi: 3.14159
Grade: A
Name: Alice
Age: 30
Pi: 3.14159
Grade: A
Name: Alice
PS C:\Users\Maori1\Desktop\TestPython> |
```

คำอธิบายเพิ่มเติม

1. ชนิดข้อมูลใน Python

- Python รองรับหลายชนิดข้อมูล เช่น:
 - int (จำนวนเต็ม): เช่น 30
 - float (จำนวนทศนิยม): เช่น 3.14159
 - str (ข้อความ): เช่น "Alice" หรือ 'A'

2. ฟังก์ชัน print

- ใช้ในการแสดงค่าของตัวแปรหรือข้อความ
- ใช้เครื่องหมายจุลภาค (,) เพื่อแยกข้อความและค่าตัวแปรที่ต้องการแสดง
- รองรับหลายชนิดข้อมูลพร้อมกันในคำสั่งเดียว

3. การใช้งาน f-strings

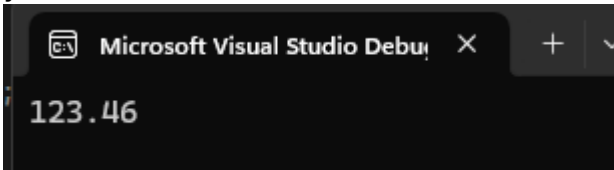
- สามารถใช้ f-string เพื่อให้โค้ดกระชับขึ้น

การแสดงผลแบบกำหนดรูปแบบ

```
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    double x = 123.456;
    cout << fixed << setprecision(2) << x << endl; // แสดงผล 2 ตำแหน่งทศนิยม
    return 0;
}
```



โปรแกรมนี้เขียนในภาษา C++ โดยมีการจัดรูปแบบการแสดงผลตัวเลขทศนิยมให้อยู่ในรูปแบบที่ต้องการ (2 ตำแหน่งทศนิยม) โดยใช้ฟังก์ชันจัดรูปแบบข้อมูลจาก <iomanip>

รายละเอียดของโปรแกรม

1. **#include <iostream>**
 - ใช้สำหรับการรับและแสดงผลข้อมูล (input/output) ผ่าน cin และ cout
2. **#include <iomanip>**
 - นำเข้าไลบรารี iomanip เพื่อใช้คำสั่งที่ช่วยจัดรูปแบบการแสดงผล เช่น fixed และ setprecision
3. **using namespace std;**
 - ช่วยให้สามารถใช้ฟังก์ชันใน namespace std ได้โดยตรง เช่น cout, endl, fixed และ setprecision โดยไม่ต้องเติม std:: ข้างหน้า
4. **ตัวแปร (Variable)**
 - **double x = 123.456;**
 - ประกาศตัวแปรชนิด double สำหรับเก็บตัวเลขทศนิยม และกำหนดค่าเริ่มต้นเป็น 123.456
5. **การจัดรูปแบบและแสดงผล**
 - **cout << fixed;**
 - ใช้ตัวเลือก fixed เพื่อกำหนดรูปแบบการแสดงผลเป็นค่าทศนิยมแบบคงที่ (fixed-point notation) แทนการใช้รูปแบบเลขยกกำลัง (scientific notation)

- `cout << setprecision(2);`
 - ใช้ฟังก์ชัน `setprecision` จาก `<iomanip>` เพื่อกำหนดจำนวนตำแหน่งทศนิยมให้แสดงเป็น 2 ตำแหน่ง
- `cout << x << endl;`
 - แสดงค่าของตัวแปร `x` ที่ถูกจัดรูปแบบเป็นทศนิยม 2 ตำแหน่ง คือ 123.46
 - `endl` ใช้สำหรับขึ้นบรรทัดใหม่หลังจากแสดงผล
- 6. `return 0;`
 - คืนค่า 0 ให้กับระบบปฏิบัติการเมื่อโปรแกรมทำงานสำเร็จ

```
x = 123.456
print("{:.2f}".format(x)) # แสดงผล 2 ตำแหน่งทศนิยม
```

```
PS C:\Users\Maori1\Desktop\TestPython> & C:/Users/Maori1/
python.exe c:/Users/Maori1/Desktop/TestPython/htllo.py
123.46
PS C:\Users\Maori1\Desktop\TestPython>
```

รายละเอียดของโค้ด

1. `x = 123.456`
 - ประกาศตัวแปร `x` และกำหนดค่าเป็น 123.456 ซึ่งเป็นตัวเลขชนิดทศนิยม (float)
2. `"{:.2f}".format(x)`
 - ใช้สตริงรูปแบบ (string formatting) เพื่อจัดรูปแบบการแสดงผลตัวเลข
 - `{:.2f}`:
 - `{}`: ใช้เป็นตัวกำหนดตำแหน่งที่จะใส่ค่าตัวแปรลงไป
 - `:.2f`:
 - `.`: หมายถึงการกำหนดตำแหน่งทศนิยม
 - `2`: ระบุจำนวนตำแหน่งทศนิยมที่ต้องการแสดง (2 ตำแหน่ง)
 - `f`: ระบุว่าต้องการแสดงในรูปแบบ ทศนิยม (floating-point)

- `.format(x)`:

- ใช้เมธอด `format` เพื่อแทนค่าของตัวแปร `x` ลงใน `{:.2f}`

3. `print("{:.2f}".format(x))`

- ฟังก์ชัน `print` ใช้สำหรับแสดงผลค่าที่ได้จากการจัดรูปแบบในบรรทัดนี้ออกทางหน้าจอ



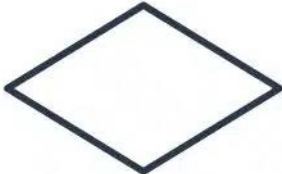

2.1.2. สัญลักษณ์ Flowchart: ภาษาภาพสำหรับอธิบายขั้นตอนการทำงาน







Flowchart หรือ แผนภูมิขั้นตอน เป็นเครื่องมือที่ใช้ในการแสดงลำดับขั้นตอนของการทำงานหรือกระบวนการต่าง ๆ โดยใช้สัญลักษณ์มาตรฐานที่เป็นที่เข้าใจกันทั่วไป สัญลักษณ์เหล่านี้จะช่วยให้เราสามารถมองเห็นภาพรวมของกระบวนการได้อย่างชัดเจน และเข้าใจถึงการเชื่อมโยงระหว่างขั้นตอนต่าง ๆ ได้ง่ายขึ้น

ทำไมต้องใช้ Flowchart?

- ช่วยให้เข้าใจกระบวนการได้ง่ายขึ้น: โดยเฉพาะอย่างยิ่งสำหรับกระบวนการที่ซับซ้อน
- ใช้ในการสื่อสาร: สามารถนำเสนอให้ผู้อื่นเข้าใจได้ง่าย
- ใช้ในการวางแผน: ช่วยให้เห็นภาพรวมของกระบวนการ และระบุจุดที่อาจเกิดปัญหาได้
- ใช้ในการแก้ปัญหา: ช่วยในการวิเคราะห์ปัญหา และหาทางแก้ไขได้อย่างมีระบบ

สัญลักษณ์ Flowchart พื้นฐาน

1. Terminator (รูปวงรี) <ul style="list-style-type: none">• ความหมาย: ใช้สำหรับจุดเริ่มต้น (Start) หรือจุดสิ้นสุด (End) ของกระบวนการ• ตัวอย่างข้อความในวงรี: "Start" หรือ "End"	 <p>Start/Stop</p>
2. Process (รูปสี่เหลี่ยมผืนผ้า) <ul style="list-style-type: none">• ความหมาย: ใช้แทนกระบวนการหรือขั้นตอนการทำงานที่ต้องปฏิบัติ• ตัวอย่างข้อความในสี่เหลี่ยม: "คำนวณยอดรวม" หรือ "ส่งอีเมล"	 <p>Process</p>
3. Decision (รูปข้าวหลามตัด) <ul style="list-style-type: none">• ความหมาย: ใช้แทนจุดที่ต้องตัดสินใจ โดยมีคำถามที่สามารถตอบได้ด้วย "ใช่ (Yes)" หรือ "ไม่ใช่ (No)"• ตัวอย่างข้อความในข้าวหลามตัด: "ข้อมูลครบถ้วนหรือไม่?"	 <p>Decision</p>
4. Arrow (ลูกศร) <ul style="list-style-type: none">• ความหมาย: ใช้เพื่อแสดงทิศทางของการทำงานหรือการเชื่อมโยงระหว่างสัญลักษณ์ต่างๆ	 <p>Flowline</p>

<ul style="list-style-type: none"> • การใช้งาน: เชื่อมต่อระหว่างวงรี, สี่เหลี่ยม, และข้าวหลามตัด เพื่อแสดงลำดับขั้นตอน 	
5. Input/Output (รูปสี่เหลี่ยมคางหมู) <ul style="list-style-type: none"> • ความหมาย: ใช้แทนการรับข้อมูล (Input) หรือแสดงผลข้อมูล (Output) • ตัวอย่างข้อความในสี่เหลี่ยมคางหมู: "ป้อนรหัสนักเรียน" หรือ "แสดงผลรวม" 	 <p>Write</p>
6. Between-Page สัญลักษณ์โฟลวชาร์ตแสดงการกำหนดจุดอ้างอิงในการเชื่อมต่อ ระหว่าง หน้ากระดาษของการเขียนผังงาน	 <p>IN-Page Connector</p>
7. Annotation สัญลักษณ์ Flowchart ที่แสดงการระบุหมายเหตุเพื่อใช้อธิบายรายละเอียดเพิ่มเติมของการดำเนินงาน	 <p>Between-Page</p>
8. Keyboard สัญลักษณ์ Flowchart แสดงถึงการรับค่าข้อมูลหรืออ่านข้อมูลเข้ามาจากคีย์บอร์ด	 <p>Read</p>
9. Monitor แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่จอภาพ(Monitor)	 <p>Monitor</p>
10. .Printer แสดงรายละเอียดข้อมูลหรือผลลัพธ์ของการประมวลผลไปที่เครื่องพิมพ์(Printer)	 <p>Printer</p>

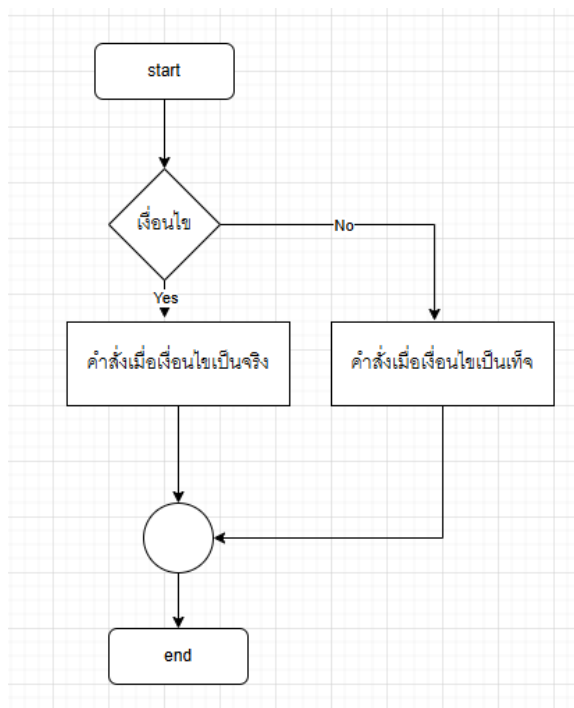
หมายเหตุ รูปนำมาจาก <https://codegeniusacademy.com/flowchart/>

2.1.3. คำสั่ง if-else, switch-case

คำสั่ง if-else

คำสั่ง **if-else** ใช้ในการตรวจสอบเงื่อนไขที่ต้องการดำเนินการในกรณีที่เงื่อนไขนั้นเป็นจริง (True) หรือไม่จริง (False) ซึ่งเหมาะสำหรับการตรวจสอบเงื่อนไขเพียง 1 ถึงหลายเงื่อนไขที่มีลำดับการตรวจสอบแบบต่อเนื่อง

โครงสร้างพื้นฐานของ if-else



C++

```
if (เงื่อนไข) {  
    // คำสั่งเมื่อเงื่อนไขเป็นจริง  
} else {  
    // คำสั่งเมื่อเงื่อนไขเป็นเท็จ  
}
```

Python

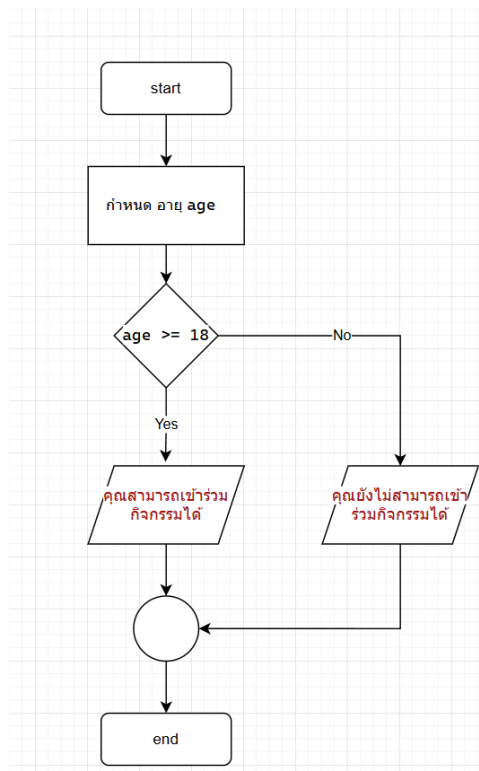
```
if เงื่อนไข:  
    # คำสั่งเมื่อเงื่อนไขเป็นจริง  
else:  
    # คำสั่งเมื่อเงื่อนไขเป็นเท็จ
```


ตัวอย่าง if-else

ตัวอย่างนี้จะทำการเปรียบเทียบค่าของตัวแปร age กับค่า 18 หากค่า age มากกว่าหรือเท่ากับ 18 จะแสดงข้อความว่าสามารถเข้าร่วมกิจกรรมได้ แต่ถ้าค่า age น้อยกว่า 18 จะแสดงข้อความว่ายังไม่สามารถเข้าร่วมกิจกรรมได้

ตัวอย่างการทำงาน:

- ถ้าเราเปลี่ยนค่าของ age เป็น 15 โปรแกรมจะแสดงผลลัพธ์ว่า "คุณยังไม่สามารถเข้าร่วมกิจกรรมได้"
- ถ้าเราเปลี่ยนค่าของ age เป็น 25 โปรแกรมจะแสดงผลลัพธ์ว่า "คุณสามารถเข้าร่วมกิจกรรมได้"



C++

```
#include <iostream>
using namespace std;

int main() {
    int age = 20;

    if (age >= 18) {
        cout << "คุณสามารถเข้าร่วมกิจกรรมได้" << endl;
    }
}
```

Python

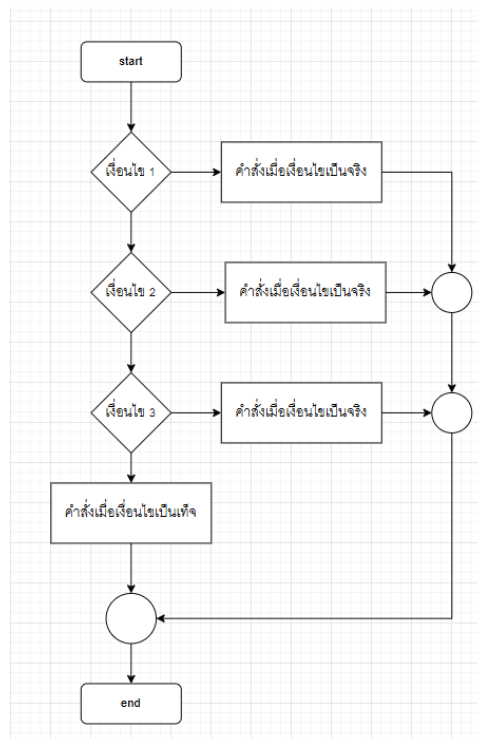
```
age = 18

if age >= 18:
    print("คุณสามารถเข้าร่วมกิจกรรมได้")
else:
    print("คุณยังไม่สามารถเข้าร่วมกิจกรรมได้")
```

<pre> } else { cout << "คุณยังไม่สามารถเข้าร่วมกิจกรรมได้" << endl; } return 0; } </pre>	
---	--

กรณีตรวจสอบหลายเงื่อนไข (if-else if-else)

เมื่อเราต้องการให้โปรแกรมตัดสินใจเลือกทำอะไรสักอย่างหนึ่งจากหลายๆ ทางเลือกตามเงื่อนไขที่แตกต่างกัน เราจะใช้โครงสร้างควบคุม if-else if-else



<pre> if (เงื่อนไขที่ 1) { // ทำงานเมื่อเงื่อนไขที่ 1 เป็นจริง } else if (เงื่อนไขที่ 2) { // ทำงานเมื่อเงื่อนไขที่ 1 เป็นเท็จ แต่เงื่อนไขที่ 2 เป็นจริง } else if (เงื่อนไขที่ 3) { // ทำงานเมื่อเงื่อนไขที่ 1 และ 2 เป็นเท็จ แต่เงื่อนไขที่ 3 เป็นจริง } </pre>	<pre> if เงื่อนไข1: # คำสั่งเมื่อเงื่อนไขเป็นจริง elif เงื่อนไข2: # คำสั่งเมื่อเงื่อนไขที่ 1 เป็นเท็จ แต่เงื่อนไขที่ 2 เป็นจริง elif เงื่อนไข3: # คำสั่งเมื่อเงื่อนไขที่ 1 และ 2 เป็นเท็จ แต่เงื่อนไขที่ 3 เป็นจริง </pre>
---	--

<pre> } else { // ทำงานเมื่อทุกเงื่อนไขเป็นเท็จ } </pre>	<pre> else: # คำสั่งเมื่อเงื่อนไขเป็นเท็จ </pre>
--	--

ตัวอย่าง if-else if-else

โปรแกรมนี้นี้มีหน้าที่ตรวจสอบเกรดนักเรียนจากคะแนนที่กำหนดไว้

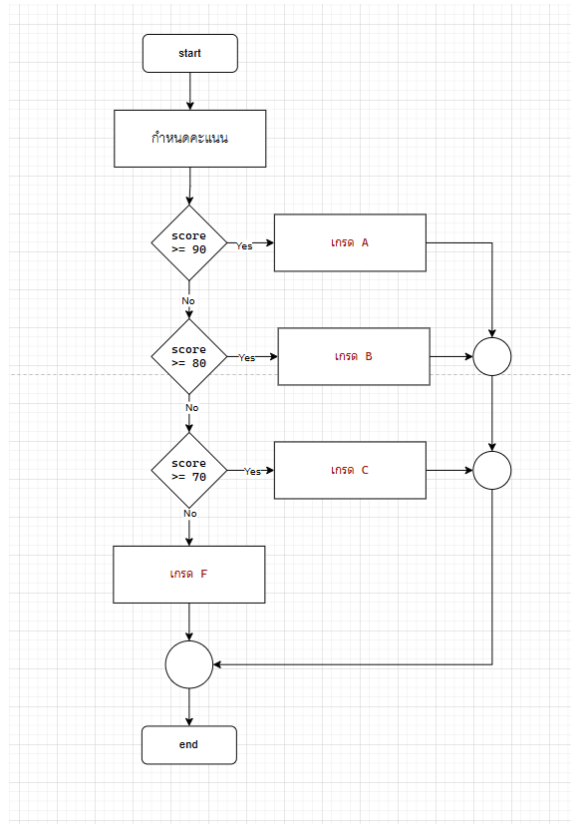
วิธีการทำงาน:

- 1) กำหนดคะแนน: โปรแกรมจะกำหนดคะแนนของนักเรียนไว้ในตัวแปร score โดยมีค่าเริ่มต้นเป็น 85 คะแนน
- 2) ตรวจสอบเงื่อนไข: โปรแกรมจะนำคะแนนไปเปรียบเทียบกับเกณฑ์ต่างๆ ที่กำหนดไว้ (90, 80, 70, 60) เพื่อหาว่าคะแนนอยู่ในช่วงของเกรดใด
- 3) แสดงผลลัพธ์: เมื่อพบเงื่อนไขที่ตรงกับคะแนน โปรแกรมจะแสดงเกรดที่สอดคล้องกันออกมาทางหน้าจอ

ตัวอย่างในกรณีนี้:

เนื่องจากคะแนน score มีค่าเท่ากับ 85 ซึ่งอยู่ในช่วง 80-89 โปรแกรมจึงแสดงผลลัพธ์ออกมาว่า "เกรด B"

สรุปง่ายๆ: โปรแกรมนี้เหมือนกับเครื่องคิดเลขเกรด ที่เราป้อนคะแนนเข้าไป แล้วโปรแกรมจะบอกให้เราทราบว่า ได้เกรดอะไร



```

#include <iostream>
using namespace std;

int main() {
    int score = 85;

    if (score >= 90) {
        cout << "เกรด A" << endl;
    }
    else if (score >= 80) {
        cout << "เกรด B" << endl;
    }
    else if (score >= 70) {
        cout << "เกรด C" << endl;
    }
    else {
        cout << "เกรด F" << endl;
    }

    return 0;
}

```

```

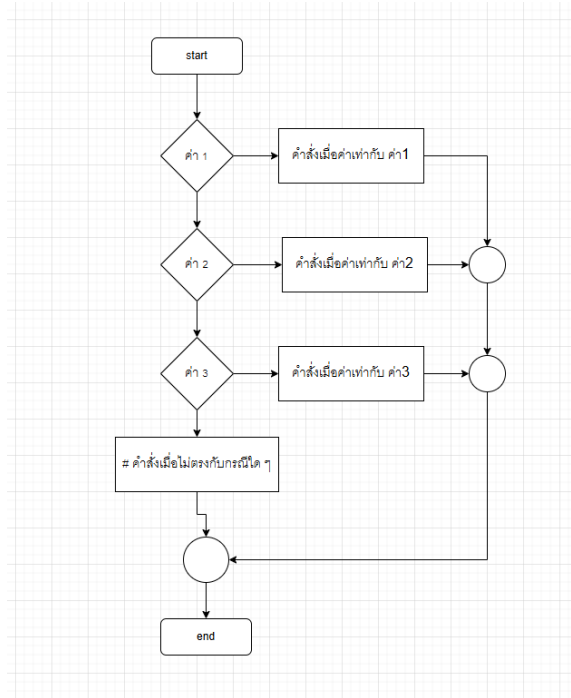
score = 85

if score >= 90:
    print("เกรด A")
elif score >= 80 :
    print("เกรด B")
elif score >= 70 :
    print("เกรด C")
else:
    print("เกรด F")

```

คำสั่ง **switch-case** (Python ใช้ **match-case** ในเวอร์ชัน 3.10 ขึ้นไป)

switch-case (หรือใน Python คือ match-case) ใช้สำหรับการตรวจสอบค่าหลายกรณีที่ชัดเจน โดยแต่ละกรณีเป็นเงื่อนไขที่มีค่าเฉพาะเจาะจง มักใช้ในกรณีที่ตัวแปรมีค่าแน่นอนจำนวนหนึ่ง เช่น ตัวเลขหรือสตริง



<pre>switch (ตัวแปร) { case ค่า1: // คำสั่งเมื่อค่าเท่ากับ ค่า1 break; case ค่า2: // คำสั่งเมื่อค่าเท่ากับ ค่า2 break; default: // คำสั่งเมื่อไม่ตรงกับกรณีใดเลย } </pre>	<pre>match ตัวแปร: case ค่า1: # คำสั่งที่ต้องการ case ค่า2: # คำสั่งที่ต้องการ case _: # คำสั่งเมื่อไม่ตรงกับกรณีใด ๆ </pre>
---	--

ตัวอย่าง

โปรแกรมนี้ถูกออกแบบมาเพื่อแสดงชื่อวันในสัปดาห์ตามตัวเลขที่กำหนดให้

วิธีการทำงาน:

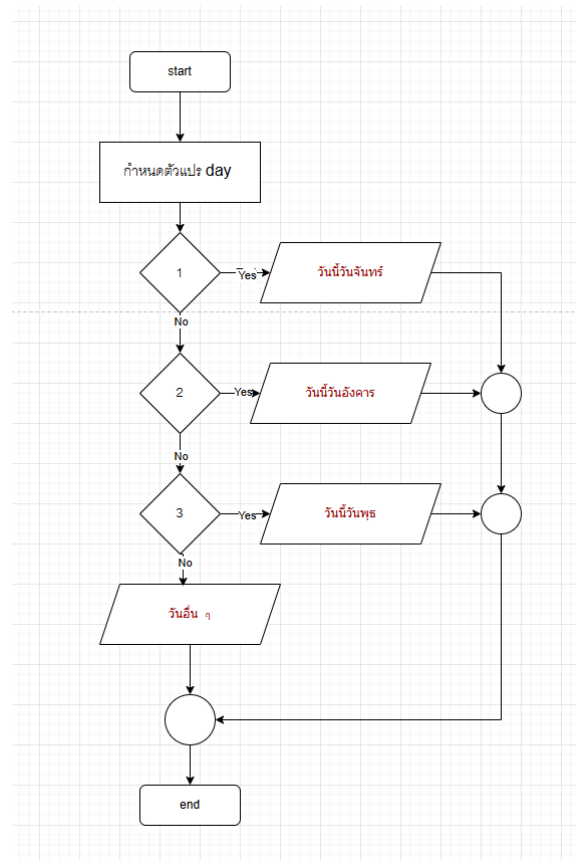
- 1) กำหนดตัวแปร: โปรแกรมจะกำหนดตัวแปร day ให้มีค่าเท่ากับ 2 ซึ่งหมายถึงวันอังคาร
- 2) ใช้คำสั่ง switch: คำสั่ง switch จะนำค่าในตัวแปร day ไปเปรียบเทียบกับค่าในแต่ละ case
- 3) ตรวจสอบเงื่อนไข:
ถ้าค่าใน day ตรงกับค่าใน case ไหน คำสั่งใน case นั้นจะถูกทำงาน
ถ้าไม่มี case ไหนตรงกับค่าใน day คำสั่งใน default จะถูกทำงาน
- 4) แสดงผลลัพธ์: เมื่อพบ case ที่ตรงกัน โปรแกรมจะแสดงข้อความที่ระบุชื่อวันนั้นๆ ออกมาทางหน้าจอ

ในกรณีนี้:

เนื่องจากค่าใน day คือ 2 คำสั่งจะตรงกับ case 2

ดังนั้นโปรแกรมจะแสดงผลลัพธ์ออกมาว่า "วันนี้วันอังคาร"

สรุปง่าย ๆ: โปรแกรมนี้เหมือนกับตารางเทียบเลขกับชื่อวันในสัปดาห์ เมื่อเราใส่เลขวันเข้าไป โปรแกรมจะบอกให้เราทราบว่าเป็นวันอะไร



<pre> #include <iostream> using namespace std; int main() { int day = 2; switch (day) { case 1: cout << "วันนี้วันจันทร์" << endl; break; case 2: cout << "วันนี้วันอังคาร" << endl; break; case 3: cout << "วันนี้วันพุธ" << endl; break; default: cout << "วันอื่น ๆ" << endl; break; } return 0; } </pre>	<pre> day = 2 match day : case 1: print("วันนี้วันจันทร์") case 2: print("วันนี้วันอังคาร") case 3: print("วันนี้วันพุธ") case _: print("วันอื่น ๆ") </pre>
--	---

เปรียบเทียบ if-else กับ switch-case		
คุณสมบัติ	if-else	switch-case /match-case
ความซับซ้อนของเงื่อนไข	ใช้ได้ทั้งเงื่อนไขที่ซับซ้อนและง่าย	เหมาะสำหรับค่าที่แน่นอน (Discrete Values)
การอ่านโค้ด	อาจซับซ้อนเมื่อมีเงื่อนไขหลายเงื่อนไข	อ่านง่ายเมื่อมีกรณีหลายกรณี
ความยืดหยุ่น	รองรับการเปรียบเทียบทุกรูปแบบ	รองรับเฉพาะการเปรียบเทียบค่าที่ชัดเจน

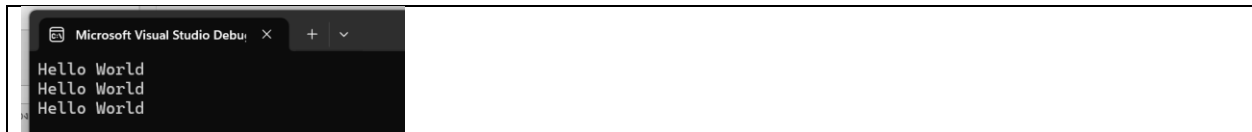
2.1.4. วนลูป for, while, do-while

ลูป (Loop) คือโครงสร้างควบคุมการทำงานของโปรแกรมที่ช่วยให้คำสั่งหรือกลุ่มคำสั่งทำงานซ้ำได้หลายครั้งโดยไม่ต้องเขียนคำสั่งเดิมซ้ำ ๆ หลายบรรทัด

ความสำคัญของลูป

1. ลดการเขียนโค้ดซ้ำ: ใช้คำสั่งที่เหมือนกันโดยไม่ต้องเขียนโค้ดซ้ำหลายครั้ง
2. ปรับเปลี่ยนตามเงื่อนไขได้: การทำงานจะหยุดเมื่อเงื่อนไขที่กำหนดสิ้นสุด
3. ลดข้อผิดพลาด: การใช้ลูปช่วยลดโอกาสเกิดข้อผิดพลาดจากการคัดลอกคำสั่งหลายครั้ง
4. ประหยัดเวลา: เขียนโปรแกรมที่ซับซ้อนและทำงานเร็วขึ้น
5. ยืดหยุ่น: ใช้กับข้อมูลจำนวนมาก เช่น การอ่านข้อมูลจากไฟล์หรือฐานข้อมูล

ตัวอย่าง:



```
Microsoft Visual Studio Debug Console
Hello World
Hello World
Hello World
```

ก่อนใช้ลูป:

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    cout << "Hello World" << endl;
    return 0;
}
```

หลังใช้ลูป:

```
#include <iostream>
using namespace std;

int main() {
    for (int i = 0; i < 3; i++) {
        cout << "Hello World" << endl;
    }
    return 0;
}
```


ประเภทของลูป

ลูปแบ่งออกเป็น 3 ประเภทหลัก:

1. For Loop

- ใช้เมื่อต้องการทำงานจำนวนรอบที่รู้ล่วงหน้า
- เช่น พิมพ์ตัวเลข 1 ถึง 10

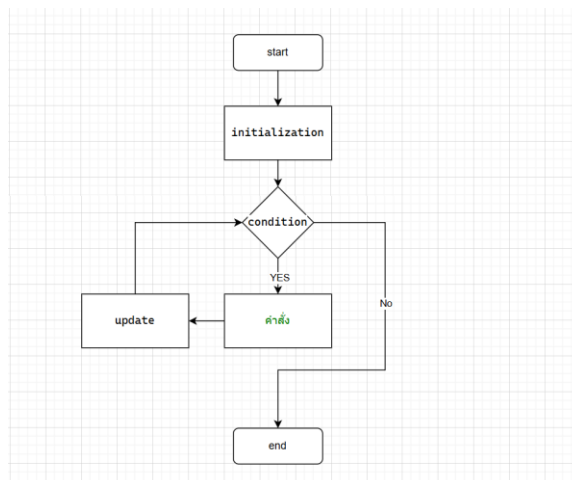
2. While Loop

- ใช้เมื่อต้องการทำงานซ้ำจนกว่าจะถึงเงื่อนไขที่กำหนด
- เช่น รอจนกว่าผู้ใช้จะป้อนค่า "exit"

3. Do-While Loop (เฉพาะใน C++)

- คล้ายกับ While แต่คำสั่งภายในจะถูกทำงานอย่างน้อยหนึ่งครั้งก่อนตรวจสอบเงื่อนไข

1. โครงสร้างของ For Loop

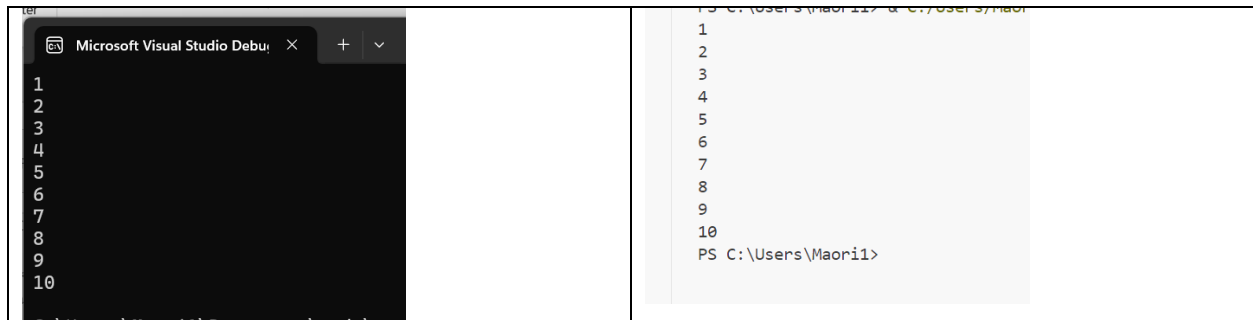


<pre>for (initialization; condition; update) { // คำสั่งที่ต้องการทำซ้ำ }</pre>	<pre>for variable in iterable: # คำสั่งที่ต้องการทำซ้ำ</pre>
---	--

คำอธิบาย

- Initialization (C++): กำหนดค่าเริ่มต้น เช่น `int i = 0;`
- Condition: เงื่อนไขที่ตรวจสอบก่อนทำซ้ำ
- Update: การปรับเปลี่ยนค่าตัวแปร เช่น `i++`

ตัวอย่าง



ก่อนใช้ลูป:

ภาษา C++	ภาษา Python
<pre>#include <iostream> using namespace std; int main() { cout << "1\n"; cout << "2\n"; cout << "3\n"; cout << "4\n"; cout << "5\n"; cout << "6\n"; cout << "7\n"; cout << "8\n"; cout << "9\n"; cout << "10\n"; return 0; }</pre>	<pre>print("1") print("2") print("3") print("4") print("5") print("6") print("7") print("8") print("9") print("10")</pre>

หลังใช้ลูป:

<pre>#include <iostream> using namespace std; int main() { for (int i = 1; i <= 10; i++) { cout << i << " "; } return 0; }</pre>	<pre>for i in range(1, 11): print(i, end=" ")</pre>
--	---

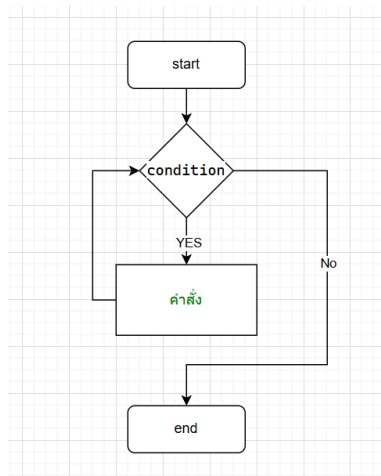
การเปรียบเทียบระหว่าง C++ และ Python

ลักษณะ	C++	Python
โครงสร้างรูป	ใช้ for(initialization; condition; update)	ใช้ for variable in iterable
เงื่อนไข	มีเงื่อนไขชัดเจนในลูป	ใช้ range() สร้างค่าอัตโนมัติ
พิมพ์ข้อความ	ใช้ cout	ใช้ print
การเว้นวรรค	ใช้ " " ใน cout	ใช้ end=" " ใน print

<p>คำอธิบายทีละบรรทัด:</p> <ul style="list-style-type: none"> - #include <iostream>: ใช้สำหรับการเรียกฟังก์ชันพิมพ์ข้อความ (cout) และรับข้อมูล (cin) - using namespace std;: เพื่อให้สามารถใช้ชื่อฟังก์ชัน เช่น cout และ cin โดยไม่ต้องเพิ่ม std:: - int main(): จุดเริ่มต้นของโปรแกรม - for (int i = 1; i <= 10; i++): <ul style="list-style-type: none"> int i = 1;: กำหนดตัวแปร i ให้มีค่าเริ่มต้นเป็น 1 i <= 10;: ทำงานจนกว่า i จะมีค่าน้อยกว่าหรือเท่ากับ 10 i++;: เพิ่มค่า i ทีละ 1 หลังจบแต่ละรอบของลูป - cout << i << " ";: พิมพ์ค่าของ i ตามด้วยช่องว่าง 	<p>for i in range(1, 11):</p> <ul style="list-style-type: none"> • ใช้คำสั่ง for เพื่อวนลูป • range(1, 11) สร้างตัวเลขตั้งแต่ 1 ถึง 10 (เลข 11 ไม่รวม) • ตัวแปร i จะมีค่าเริ่มต้นจาก 1 และเพิ่มทีละ 1 จนถึง 10 <p>print(i, end=" "):</p> <ul style="list-style-type: none"> • พิมพ์ค่าของ i • end=" " ใช้เพื่อเว้นวรรคแทนการขึ้นบรรทัดใหม่
---	---

- return 0; คืนค่ากลับสู่ระบบปฏิบัติการเพื่อ บอกว่าโปรแกรมจบการทำงานสำเร็จ	
---	--

2 โครงสร้างของ While Loop



<pre>while (เงื่อนไข) { // โค้ดที่ต้องการทำซ้ำ }</pre>	<pre>while เงื่อนไข: # โค้ดที่ต้องการทำซ้ำ</pre>
--	--

คำอธิบาย

- Condition: เงื่อนไขที่ตรวจสอบก่อนเริ่มทำงาน
- ทำงานซ้ำจนกว่าจะไม่เป็นจริง

ตัวอย่าง แสดงตัวเลข 1-10

<pre>#include <iostream> using namespace std; int main() { int i = 1; while (i <= 10) { cout << i << " \n"; i++; } return 0; }</pre>	<pre>i = 1 while i <= 10: print(i, end="\n") i += 1</pre>
--	--

3. Do-While Loop (เฉพาะใน C++)

โครงสร้างของ Do-While Loop

<pre>do { // คำสั่งที่ต้องการทำซ้ำ } while (condition);</pre>

คำอธิบาย

- คำสั่งภายในบล็อกจะทำงานอย่างน้อย 1 ครั้งก่อนตรวจสอบเงื่อนไข
- ใช้ในกรณีที่ต้องการให้คำสั่งรันก่อนตรวจสอบเงื่อนไข

ตัวอย่าง

C++: รับค่าอินพุตจากผู้ใช้จนกว่าจะป้อนตัวเลขบวก

 <pre>Microsoft Visual Studio Debug Console Enter a positive number: -5 Enter a positive number: -7 Enter a positive number: -10 Enter a positive number: 1 You entered: 1</pre>

```
#include <iostream>
using namespace std;

int main() {
    int num;
    do {
        cout << "Enter a positive number: ";
        cin >> num;
    } while (num <= 0);

    cout << "You entered: " << num;
    return 0;
}
```

ความแตกต่างระหว่าง C++ และ Python ในการเขียนลูป

ลักษณะ	C++	Python
โครงสร้าง For	ใช้ for(initialization; condition; update)	ใช้ for variable in iterable
วงเล็บปีกกา	ต้องใช้ {}	ใช้การเยื้อง (Indentation)
Do-While Loop	มีให้ใช้งาน	ไม่มีใน Python
คำสั่งเสริม (break/continue)	มีเหมือนกัน	มีเหมือนกัน

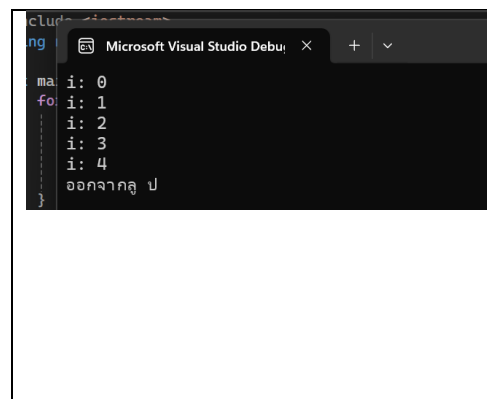
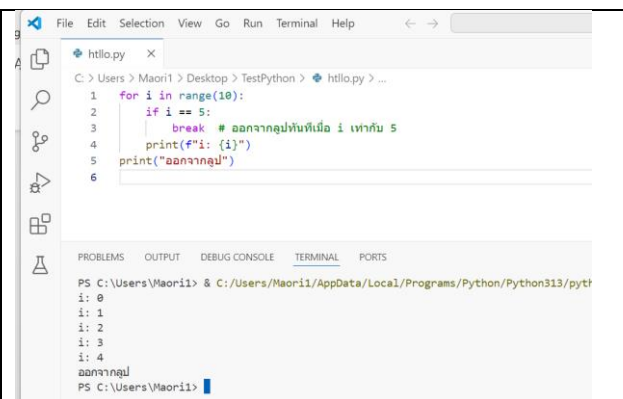
คำสั่ง break และ continue ในการวนลูปเป็นคำสั่งที่ช่วยควบคุมการทำงานของลูปในโปรแกรมได้อย่างมีประสิทธิภาพ ทั้งใน C++ และ Python มีการใช้งานที่คล้ายกันดังนี้:

คำสั่ง break

- ใช้เพื่อออกจากลูปก่อนที่จะครบเงื่อนไขของการวนซ้ำ
- เมื่อโปรแกรมเจอคำสั่ง break ในลูป จะหยุดการวนซ้ำทันทีและออกไปยังคำสั่งถัดไปหลังลูป

ตัวอย่างการใช้ใน C++

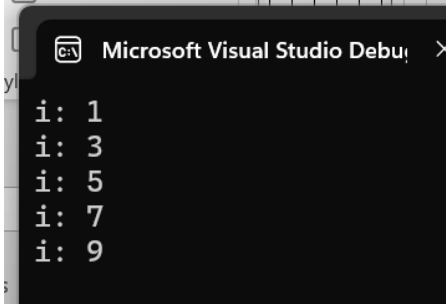
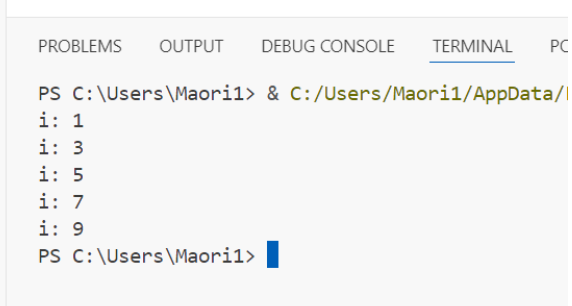
<pre>#include <iostream> using namespace std; int main() { for (int i = 0; i < 10; i++) { if (i == 5) { break; // ออกจากลูปทันทีเมื่อ i เท่ากับ 5 } cout << "i: " << i << endl; } cout << "ออกจากลูป" << endl; return 0; }</pre>	<pre>for i in range(10): if i == 5: break # ออกจากลูปทันทีเมื่อ i เท่ากับ 5 print(f"i: {i}") print("ออกจากลูป")</pre>
--	---

	
---	--

คำสั่ง continue

- ใช้เพื่อข้ามการทำงานในรอบนั้น และวนลูปต่อไปยังรอบถัดไปทันที
- ไม่ออกจากลูป แต่ข้ามคำสั่งที่อยู่หลัง continue

<pre>#include <iostream> using namespace std; int main() { for (int i = 0; i < 10; i++) { if (i % 2 == 0) { continue; // ข้ามการทำงานในรอบที่ i } cout << "i: " << i << endl; } return 0; }</pre>	<pre>for i in range(10): if i % 2 == 0: continue # ข้ามการทำงานในรอบที่ i เป็นเลขคู่ print(f"i: {i}")</pre>
---	---

	
---	--

สรุป

- **break:** หยุดการทำงานของลูปทันที
- **continue:** ข้ามการทำงานในรอบนั้น แล้วไปยังรอบถัดไป
- การใช้งานใน C++ และ Python มีความคล้ายคลึงกันมาก โดยโครงสร้างลูปเช่น for และ while รองรับคำสั่งเหล่านี้ทั้งคู่

2.1.5. โจทย์: ระบบควบคุมอุณหภูมิในห้องปฏิบัติการ Temperature Control System

คำอธิบาย:

ห้องปฏิบัติการต้องการรักษาอุณหภูมิให้อยู่ในช่วง 20-25 องศาเซลเซียสเพื่อให้แน่ใจว่าอุปกรณ์และวัตถุดิบไม่เสียหาย ระบบควบคุมอุณหภูมิจะตรวจสอบอุณหภูมิทุกๆ 5 วินาทีและทำการปรับอุณหภูมิตามเงื่อนไขต่อไปนี้:

- 1) ถ้าอุณหภูมิต่ำกว่า 20 องศาเซลเซียส ให้เปิดเครื่องทำความร้อน
- 2) ถ้าอุณหภูมิสูงกว่า 25 องศาเซลเซียส ให้เปิดเครื่องทำความเย็น
- 3) ถ้าอุณหภูมิอยู่ในช่วง 20-25 องศาเซลเซียส ให้ปิดทั้งเครื่องทำความร้อนและเครื่องทำความเย็น

1) วัตถุประสงค์ของการเขียนโปรแกรม

พัฒนาโปรแกรมที่สามารถตรวจสอบและควบคุมอุณหภูมิของห้องปฏิบัติการให้อยู่ในช่วง 20-25 องศาเซลเซียส

ป้องกันไม่ให้อุปกรณ์และวัตถุดิบในห้องปฏิบัติการเสียหายเนื่องจากอุณหภูมิที่ไม่เหมาะสม

แสดงสถานะการทำงานของระบบ (เปิดเครื่องทำความร้อน, เปิดเครื่องทำความเย็น, หรือปิดทั้งสอง)

2) รูปแบบผลลัพธ์ที่ต้องการ

โปรแกรมจะแสดงข้อความแสดงสถานะของระบบทุก 5 วินาที เช่น:

"อุณหภูมิ: 18°C -> เปิดเครื่องทำความร้อน"

"อุณหภูมิ: 26°C -> เปิดเครื่องทำความเย็น"

"อุณหภูมิ: 22°C -> ปิดเครื่องทำความร้อนและเครื่องทำความเย็น"

3) ข้อมูลนำเข้า

- อุณหภูมิปัจจุบัน (Current Temperature): อ่านค่าจากเซ็นเซอร์หรือป้อนค่าจำลอง (ในกรณีทดสอบโปรแกรม)

4) ตัวแปรที่ใช้

- C++:

- float currentTemperature: เก็บค่าอุณหภูมิที่ตรวจสอบ
- string action: เก็บข้อความแสดงสถานะของระบบ

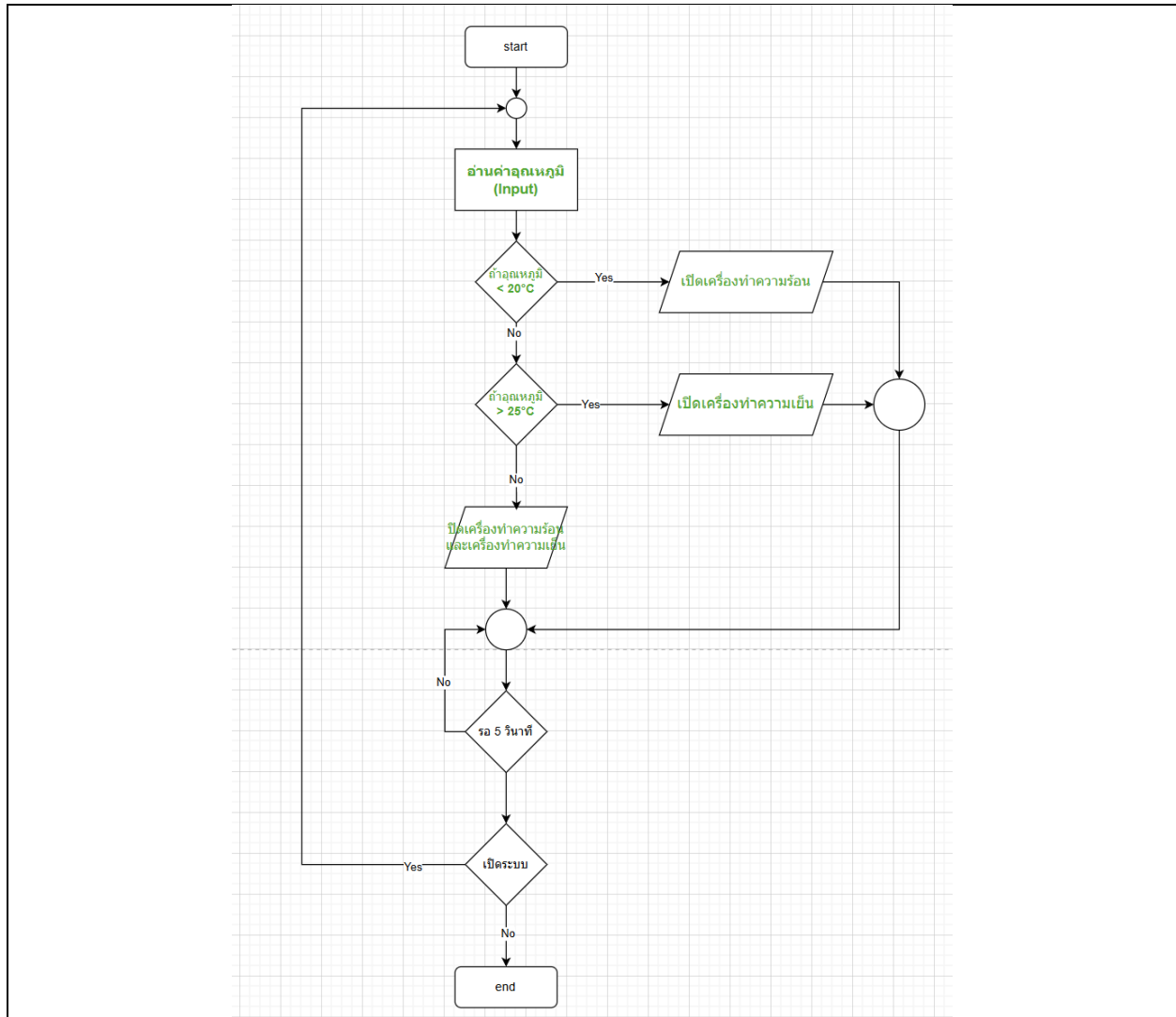
- Python:

- current_temperature: เก็บค่าอุณหภูมิที่ตรวจสอบ
- action: เก็บข้อความแสดงสถานะของระบบ

5) วิธีการประมวลผล

1. อ่านค่าอุณหภูมิ (Input): อ่านค่าจากเซ็นเซอร์ (หรือจำลองค่า)
2. ตรวจสอบเงื่อนไข (Processing):
 - ถ้าอุณหภูมิ $< 20^{\circ}\text{C}$: แสดงข้อความ "เปิดเครื่องทำความร้อน"
 - ถ้าอุณหภูมิ $> 25^{\circ}\text{C}$: แสดงข้อความ "เปิดเครื่องทำความเย็น"
 - ถ้าอุณหภูมิ อยู่ในช่วง $20-25^{\circ}\text{C}$: แสดงข้อความ "ปิดเครื่องทำความร้อนและเครื่องทำความเย็น"
3. แสดงสถานะ (Output): แสดงข้อความสถานะทุก 5 วินาที
4. วงซ้ำ: ดำเนินการซ้ำทุก 5 วินาที

6) วาด Flowchart



7) เขียนโปรแกรม

```
#include <iostream>
#include <thread> // สำหรับการหน่วงเวลา
#include <chrono> // สำหรับการหน่วงเวลา
using namespace std;

int main() {
    float currentTemperature;
    char choice; // ตัวแปรสำหรับเก็บตัวเลือกของผู้ใช้

    while (true) {
        // ถามผู้ใช้งานว่าจะใส่อุณหภูมิหรือไม่
        cout << "Do you want to enter the temperature? (Y/N): ";
        cin >> choice;
```

```

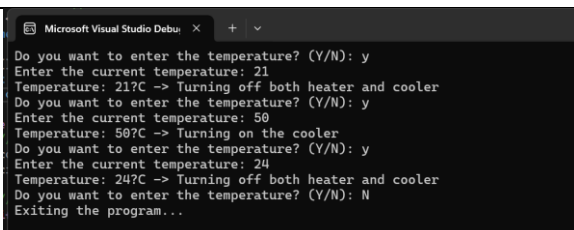
// ตรวจสอบคำตอบของผู้ใช้
if (choice == 'Y' || choice == 'y') {
    // รับค่าอุณหภูมิจากผู้ใช้
    cout << "Enter the current temperature: ";
    cin >> currentTemperature;

    // ตรวจสอบเงื่อนไข
    if (currentTemperature < 20) {
        cout << "Temperature: " << currentTemperature << "°C -> Turning on the
heater" << endl;
    }
    else if (currentTemperature > 25) {
        cout << "Temperature: " << currentTemperature << "°C -> Turning on the
cooler" << endl;
    }
    else {
        cout << "Temperature: " << currentTemperature << "°C -> Turning off
both heater and cooler" << endl;
    }

    // นอนเวลา 5 วินาที
    this_thread::sleep_for(chrono::seconds(5));
}
else {
    // ถ้าผู้ใช้ตอบ N หรือค่าที่ไม่ใช่ Y ให้หยุดโปรแกรม
    cout << "Exiting the program..." << endl;
    break;
}
}

return 0;
}

```



```

Microsoft Visual Studio Debu  X  +  v
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 21
Temperature: 21°C -> Turning off both heater and cooler
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 50
Temperature: 50°C -> Turning on the cooler
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 24
Temperature: 24°C -> Turning off both heater and cooler
Do you want to enter the temperature? (Y/N): N
Exiting the program...

```

คำอธิบายโค้ด

1. เพิ่มตัวเลือกให้ผู้ใช้:
 - ใช้ตัวแปร char choice สำหรับเก็บตัวเลือกที่ผู้ใช้ป้อน (Y สำหรับใส่อุณหภูมิต่อ, N หรืออื่น ๆ สำหรับหยุดโปรแกรม)
 - ถามผู้ใช้ในทุกการวนลูปว่า "Do you want to enter the temperature? (Y/N):"
2. ตรวจสอบคำตอบ:
 - ถ้าผู้ใช้ป้อน 'Y' หรือ 'y':

- โปรแกรมจะรับค่าอุณหภูมิและดำเนินการตามเงื่อนไข
- ถ้าผู้ใช้ป้อน 'N' หรือค่าอื่น ๆ:
 - โปรแกรมจะแสดงข้อความ "Exiting the program..." และออกจากลูปด้วยคำสั่ง break

3. การทำงานปกติ:

- เมื่อผู้ใช้เลือกใส่อุณหภูมิ โปรแกรมจะตรวจสอบอุณหภูมิที่ป้อนและแสดงผลลัพธ์ตามเงื่อนไข (เช่น เปิด/ปิดเครื่องทำความร้อนหรือทำความเย็น)

4. การหยุดโปรแกรม:

- หากผู้ใช้เลือก 'N' หรือค่าอื่น ๆ โปรแกรมจะหยุดการทำงานทันที

```
import time # สำหรับการหน่วงเวลา

def main():
    while True:
        # ถามผู้ใช้งานว่าจะใส่อุณหภูมิหรือหยุดโปรแกรม
        choice = input("Do you want to enter the temperature? (Y/N): ").strip().lower()

        # ตรวจสอบคำตอบของผู้ใช้
        if choice == 'y':
            try:
                # รับค่าอุณหภูมิจากผู้ใช้
                current_temperature = float(input("Enter the current temperature: "))
            except ValueError:
                print("Please enter a valid number!")
                continue

            # ตรวจสอบเงื่อนไข
            if current_temperature < 20:
                print(f"Temperature: {current_temperature}°C -> Turning on the heater")
            elif current_temperature > 25:
                print(f"Temperature: {current_temperature}°C -> Turning on the cooler")
            else:
                print(f"Temperature: {current_temperature}°C -> Turning off both heater and cooler")
```

```

        # หน่วงเวลา 5 วินาทีก่อนวนลูปใหม่
        time.sleep(5)

    else:
        # ถ้าผู้ใช้ตอบ N หรือค่าอื่น ๆ ให้หยุดโปรแกรม
        print("Exiting the program...")
        break

if __name__ == "__main__":
    main()

```

```

i: 7
i: 9
PS C:\Users\Maori1> & C:/Users/Maori1/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Maori1/Desktop/TestPython/htllo.py
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 21
Temperature: 21.0°C -> Turning off both heater and cooler
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 35
Temperature: 35.0°C -> Turning on the cooler
Do you want to enter the temperature? (Y/N): y
Enter the current temperature: 24
Temperature: 24.0°C -> Turning off both heater and cooler
Do you want to enter the temperature? (Y/N): n
Exiting the program...
PS C:\Users\Maori1>

```

โค้ด Python ที่ปรับปรุง

python

Copy code

import time # สำหรับการหน่วงเวลา

def main():

while True:

ถามผู้ใช่ว่าจะใส่อุณหภูมิหรือหยุดโปรแกรม

choice = input("Do you want to enter the temperature? (Y/N): ").strip().lower()

ตรวจสอบคำตอบของผู้ใช้

if choice == 'y':

try:

รับค่าอุณหภูมิจากผู้ใช้

```

        current_temperature = float(input("Enter the current temperature: "))
    except ValueError:
        print("Please enter a valid number!")
        continue

    # ตรวจสอบเงื่อนไข
    if current_temperature < 20:
        print(f"Temperature: {current_temperature}°C -> Turning on the heater")
    elif current_temperature > 25:
        print(f"Temperature: {current_temperature}°C -> Turning on the cooler")
    else:
        print(f"Temperature: {current_temperature}°C -> Turning off both heater and
cooler")

    # หน่วงเวลา 5 วินาทีก่อนวนลูปใหม่
    time.sleep(5)

else:
    # ถ้าผู้ใช้ตอบ N หรือค่าอื่น ๆ ให้หยุดโปรแกรม
    print("Exiting the program...")
    break

if __name__ == "__main__":
    main()

```

คำอธิบายโค้ด

1. ถามผู้ใช้:

- ใช้ input() เพื่อถามผู้ใช้ว่า "Do you want to enter the temperature? (Y/N):"
- คำตอบของผู้ใช้ถูกเปลี่ยนเป็นตัวพิมพ์เล็กด้วย .lower() เพื่อรองรับทั้ง Y และ y.

2. ตรวจสอบคำตอบ:

- ถ้าผู้ใช้ป้อน 'y':
 - โปรแกรมจะรับค่าอุณหภูมิ และดำเนินการตามเงื่อนไข
- ถ้าผู้ใช้ป้อน 'n' หรืออื่น ๆ:
 - โปรแกรมจะแสดงข้อความ "Exiting the program..." และหยุดการทำงานด้วย break.

3. ตรวจสอบอุณหภูมิ:

- ใช้เงื่อนไข if, elif, และ else:
 - ถ้าอุณหภูมิต่ำกว่า 20°C: แสดงข้อความ "Turning on the heater"
 - ถ้าอุณหภูมิสูงกว่า 25°C: แสดงข้อความ "Turning on the cooler"
 - ถ้าอยู่ระหว่าง 20-25°C: แสดงข้อความ "Turning off both heater and cooler"

4. การหน่วงเวลา:

- ใช้ time.sleep(5) เพื่อหน่วงเวลา 5 วินาทีก่อนเริ่มวนลูปใหม่

5. จัดการข้อผิดพลาด (Error Handling):

- ใช้ try-except เพื่อจัดการกรณีที่ผู้ใช้ป้อนค่าที่ไม่ใช่ตัวเลข (ป้องกันโปรแกรมล่ม)

2.1.6. โจทย์: ระบบควบคุมแขนหุ่นยนต์ในการจับวัตถุ

คำอธิบาย:

แขนหุ่นยนต์ในโรงงานผลิตต้องการจับวัตถุจากแท่นส่ง (conveyor belt) และวางลงในกล่อง แขนหุ่นยนต์มีเซนเซอร์สำหรับตรวจจับวัตถุและตำแหน่งของวัตถุ ระบบควบคุมจะตรวจสอบตำแหน่งของวัตถุและทำการควบคุมแขนหุ่นยนต์ให้เคลื่อนไหวยตามเงื่อนไขต่อไปนี้:

1. ถ้าวัตถุอยู่ในตำแหน่ง A ให้แขนหุ่นยนต์เคลื่อนไหวยไปทางซ้าย
2. ถ้าวัตถุอยู่ในตำแหน่ง B ให้แขนหุ่นยนต์เคลื่อนไหวยไปทางขวา
3. ถ้าวัตถุอยู่ในตำแหน่ง C ให้แขนหุ่นยนต์เคลื่อนไหวยลงล่างและจับวัตถุ
4. ถ้าไม่มีวัตถุ ให้แขนหุ่นยนต์กลับไปยังตำแหน่งเริ่มต้น

การวิเคราะห์งานระบบควบคุมแขนหุ่นยนต์

1) วัตถุประสงค์ของการเขียนโปรแกรม

- พัฒนาโปรแกรมควบคุมการเคลื่อนไหวยของแขนหุ่นยนต์เพื่อจับวัตถุจากสายพานลำเลียงและวางลงในกล่อง
- กำหนดพฤติกรรมของแขนหุ่นยนต์ตามตำแหน่งของวัตถุ (A, B, C หรือไม่มีวัตถุ)
- มีฟังก์ชันเพื่อถามผู้ใช้งานที่ต้องการให้ระบบทำงานต่อหรือหยุด

2) รูปแบบผลลัพธ์ที่ต้องการ

- โปรแกรมแสดงข้อความระบุการเคลื่อนไหวยของแขนหุ่นยนต์:
 - ตำแหน่ง A: "Moving arm to the left."
 - ตำแหน่ง B: "Moving arm to the right."
 - ตำแหน่ง C: "Moving arm down to pick up the object."
 - ไม่มีวัตถุ: "Returning arm to the starting position."

- มีการถามผู้ใช่ว่า "Do you want to continue? (Y/N):"

3) ข้อมูลนำเข้า

- ตำแหน่งของวัตถุ: A, B, C, หรือไม่มีวัตถุ
- ตัวเลือกจากผู้ใช้: Y (ทำงานต่อ) หรือ N (หยุดโปรแกรม)

4) ตัวแปรที่ใช้

- C++:
 - char position: เก็บตำแหน่งของวัตถุ (A, B, C หรือ 'N' สำหรับไม่มีวัตถุ)
 - char choice: เก็บตัวเลือกของผู้ใช้ (Y หรือ N)
- Python:
 - position: เก็บตำแหน่งของวัตถุ (A, B, C หรือ 'N')
 - choice: เก็บตัวเลือกของผู้ใช้ (Y หรือ N)

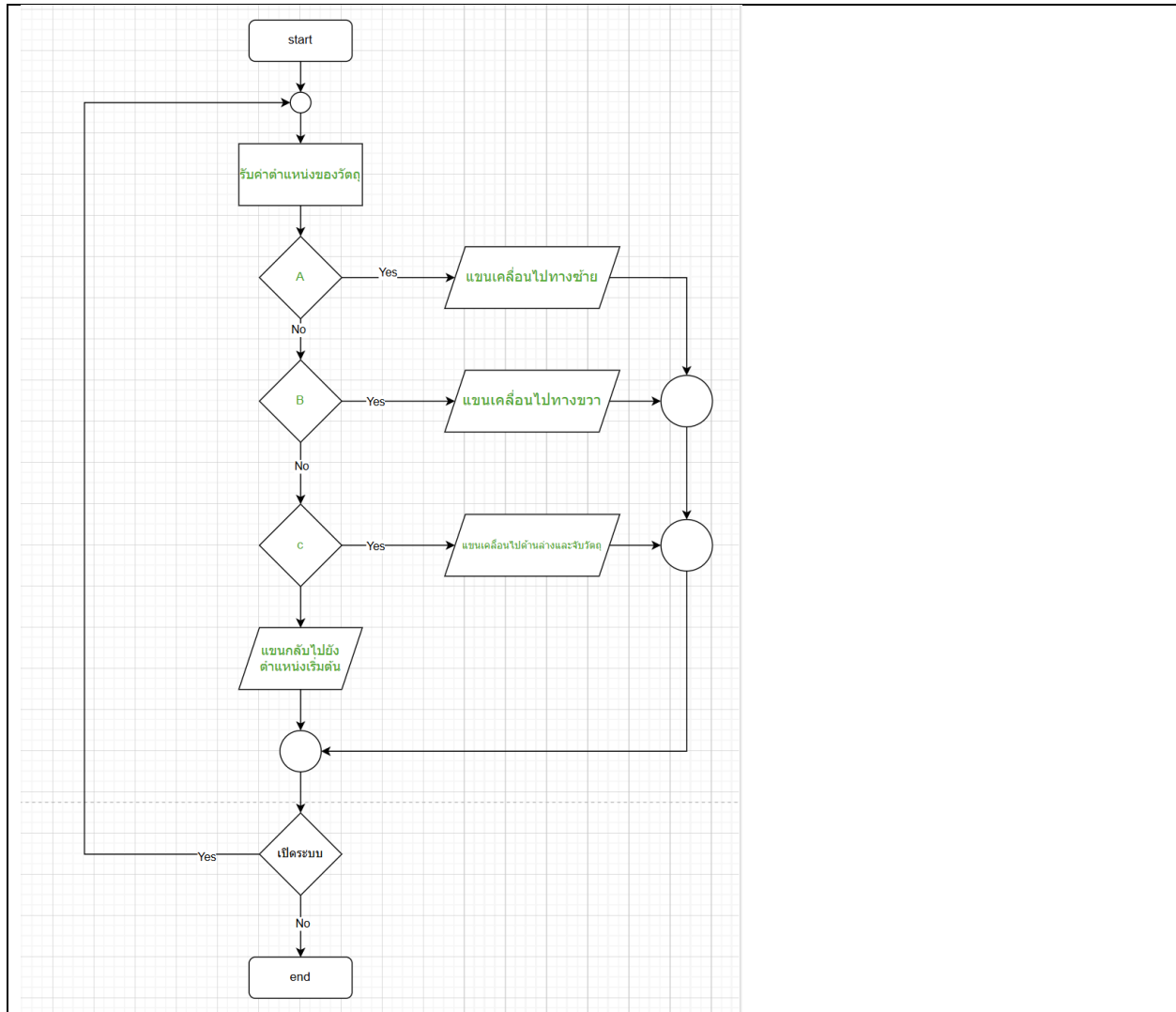
5) วิธีการประมวลผล

1. รับค่าตำแหน่งของวัตถุจากผู้ใช้
2. ตรวจสอบตำแหน่งของวัตถุ:
 - ตำแหน่ง A: แขนเคลื่อนไปทางซ้าย
 - ตำแหน่ง B: แขนเคลื่อนไปทางขวา
 - ตำแหน่ง C: แขนเคลื่อนไปด้านล่างและจับวัตถุ
 - ไม่มีวัตถุ: แขนกลับไปยังตำแหน่งเริ่มต้น
3. แสดงการกระทำของแขนหุ่นยนต์

4. ถามผู้ใช้งานว่าจะทำงานต่อหรือไม่:

- ถ้าตอบ Y: กลับไปเริ่มต้นใหม่
- ถ้าตอบ N: หยุดโปรแกรม

6) FlowChart



7.เขียนโปรแกรม

```
#include <iostream>
using namespace std;

int main() {
    char position;
    char choice;

    do {
        // รับตำแหน่งของวัตถุ
        cout << "Enter the position of the object (A, B, C, or N for no object): ";
        cin >> position;

        // ตรวจสอบตำแหน่งและแสดงการกระทำของแขนหุ่นยนต์
        switch (position) {
            case 'A':
            case 'a':
                cout << "Moving arm to the left." << endl;
                break;
            case 'B':
            case 'b':
                cout << "Moving arm to the right." << endl;
                break;
            case 'C':
            case 'c':
                cout << "Moving arm down to pick up the object." << endl;
                break;
            case 'N':
            case 'n':
                cout << "Returning arm to the starting position." << endl;
                break;
            default:
                cout << "Invalid position entered!" << endl;
        }

        // ถามผู้ใช้งานว่าจะทำงานต่อหรือไม่
        cout << "Do you want to continue? (Y/N): ";
        cin >> choice;

    } while (choice == 'Y' || choice == 'y'); // ทำงานต่อเมื่อผู้ใช้ตอบ Y

    cout << "Exiting the program..." << endl;
    return 0;
}
```

```

def main():
    while True:
        # รับตำแหน่งของวัตถุจากผู้ใช้
        position = input("Enter the position of the object (A, B, C, or N for
no object): ").strip().upper()

        # ตรวจสอบตำแหน่งและแสดงการกระทำของแขนหุ่นยนต์
        if position == 'A':
            print("Moving arm to the left.")
        elif position == 'B':
            print("Moving arm to the right.")
        elif position == 'C':
            print("Moving arm down to pick up the object.")
        elif position == 'N':
            print("Returning arm to the starting position.")
        else:
            print("Invalid position entered!")

        # ถามผู้ใช่ว่าจะทำงานต่อหรือไม่
        choice = input("Do you want to continue? (Y/N): ").strip().upper()
        if choice != 'Y':
            print("Exiting the program...")
            break

if __name__ == "__main__":
    main()

```

คำอธิบายโค้ด

1. รับตำแหน่งของวัตถุ:
 - ใน C++ ใช้ cin รับค่าตำแหน่ง และใน Python ใช้ input() โดยแปลงให้เป็นตัวพิมพ์ใหญ่เพื่อรองรับทั้งพิมพ์เล็กและพิมพ์ใหญ่
2. ตรวจสอบตำแหน่งวัตถุ:
 - ใช้ switch ใน C++ และ if-elif ใน Python เพื่อตรวจสอบค่าตำแหน่ง (A, B, C หรือ N)
 - แสดงข้อความตามตำแหน่งที่กำหนด
3. ถามผู้ใช่ว่าจะทำงานต่อหรือไม่:
 - ใน C++ ใช้ cin เพื่อถามผู้ใช่ว่าต้องการทำงานต่อหรือไม่ (Y/N)

- ใน Python ใช้ input() และตรวจสอบเงื่อนไขเพื่อหยุดหรือวนลูป

4. หยุดโปรแกรม:

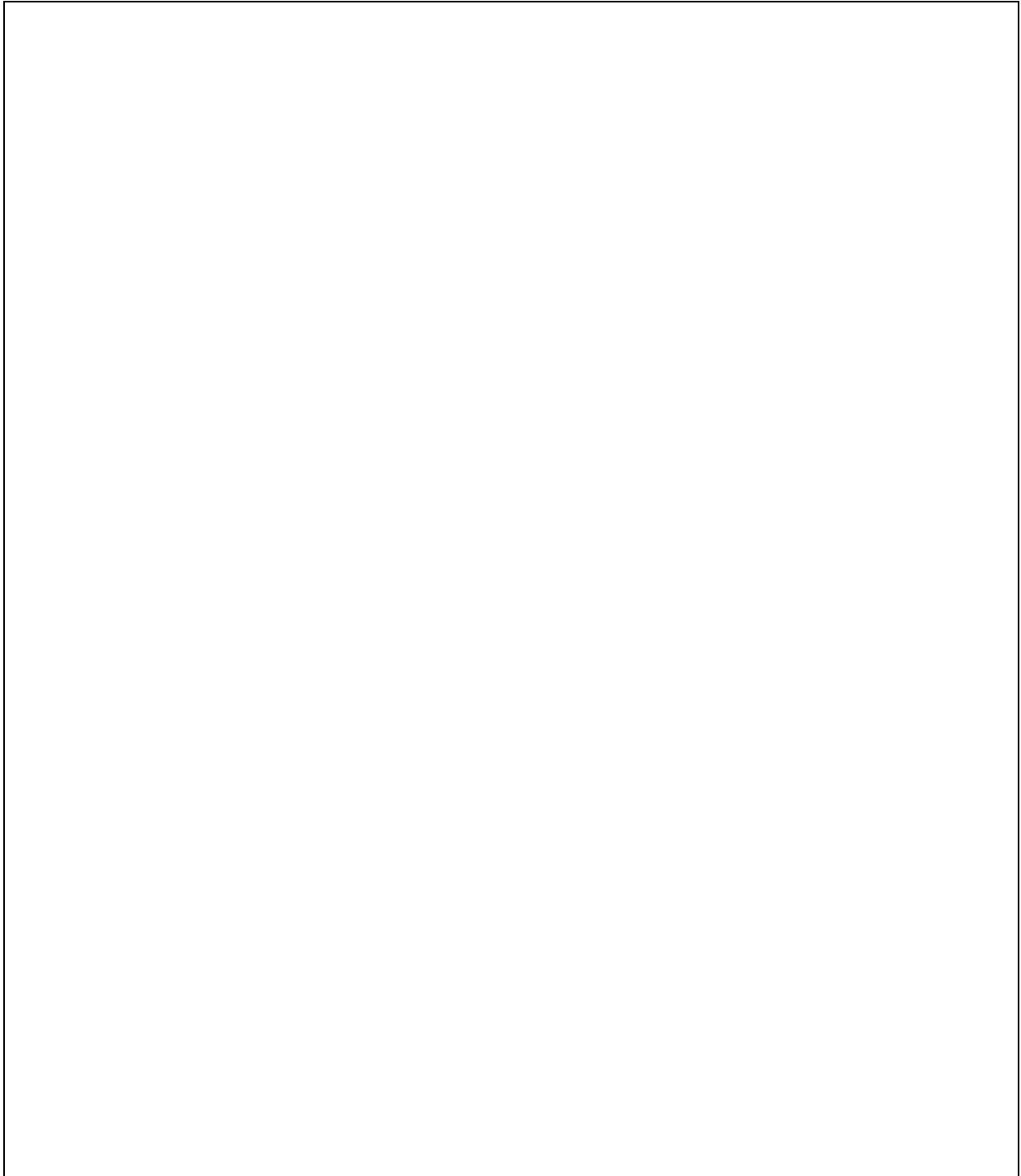
- เมื่อผู้ใช้ตอบ N หรือค่าอื่นที่ไม่ใช่ Y โปรแกรมจะแสดงข้อความและหยุดการทำงาน

```
Microsoft Visual Studio Debu...  
Enter the position of the object (A, B, C, or N for no object): A  
Moving arm to the left.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): N  
Returning arm to the starting position.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): B  
Moving arm to the right.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): C  
Moving arm down to pick up the object.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): F  
Invalid position entered!  
Do you want to continue? (Y/N): N  
Exiting the program...
```

```
PS C:\Users\Maorii> & C:/Users/Maorii/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Maorii/Desktop/TestPython/htllo.py  
Enter the position of the object (A, B, C, or N for no object): A  
Moving arm to the left.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): B  
Moving arm to the right.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): C  
Moving arm down to pick up the object.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): N  
Returning arm to the starting position.  
Do you want to continue? (Y/N): Y  
Enter the position of the object (A, B, C, or N for no object): Z  
Invalid position entered!  
Do you want to continue? (Y/N): N  
Exiting the program...  
PS C:\Users\Maorii> |
```

2.1.7. พิมพ์เลข 1 ถึง 10 ด้วย for, while, และ do-while

วาด Flowchart



2.1.8. ระบบควบคุมรถวิ่งตามเส้น (Line Following Robot)

คำอธิบาย:

รถวิ่งตามเส้นต้องการวิ่งตามเส้นสีดำบนพื้นสีขาว รถมีเซนเซอร์สี (color sensor) ที่ติดตั้งที่ด้านหน้าของรถ เซนเซอร์จะตรวจจับสีและส่งข้อมูลกลับมาว่าเส้นสีดำอยู่ทางซ้าย, ขวา, หรือกลาง ระบบควบคุมจะตรวจสอบข้อมูลจากเซนเซอร์และทำการควบคุมรถให้เคลื่อนไหวตามเงื่อนไขต่อไปนี้:

- 1) ถ้าเส้นสีดำอยู่ทางซ้าย ให้รถเลี้ยวไปทางซ้าย
- 2) ถ้าเส้นสีดำอยู่ทางขวา ให้รถเลี้ยวไปทางขวา
- 3) ถ้าเส้นสีดำอยู่กลาง ให้รถวิ่งตรงไปข้างหน้า

วาด Flowchart



2.1.9. การใช้ฟังก์ชันร่วมกับคำสั่งควบคุมการทำงาน

2.2. ปฏิบัติ (3 ชม.)

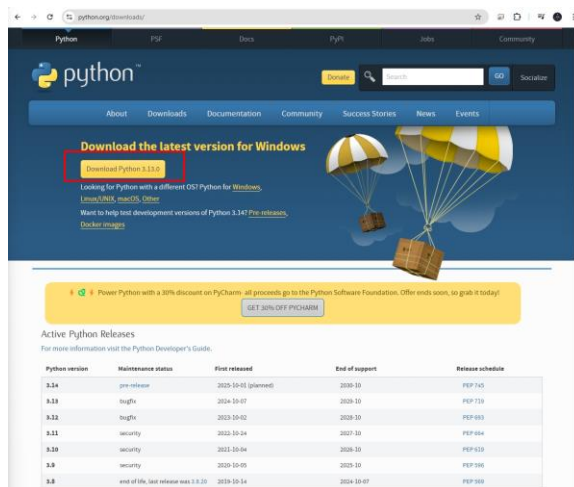
1.1.1. การติดตั้ง Visual Studio Code และ Python: คู่มือฉบับละเอียดสำหรับผู้เริ่มต้น

Visual Studio Code (VS Code) เป็นโปรแกรมแก้ไขโค้ดที่ได้รับความนิยมอย่างมาก เนื่องจากใช้งานง่าย มีส่วนเสริมมากมาย และรองรับภาษาโปรแกรมหลากหลาย รวมถึง **Python** ซึ่งเป็นภาษาโปรแกรมที่ได้รับความนิยมอย่างแพร่หลายในปัจจุบัน

ขั้นตอนการติดตั้ง

1. ดาวน์โหลด์และติดตั้ง Python:

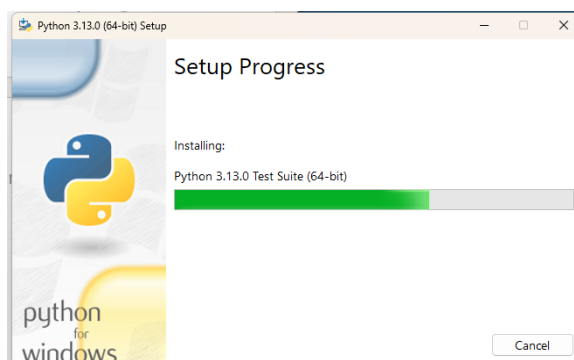
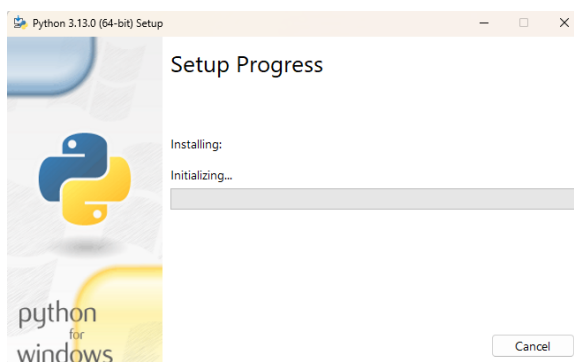
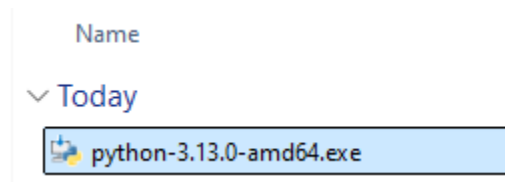
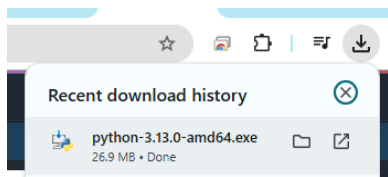
- เข้าเว็บไซต์ **Python**: ไปที่ <https://www.python.org/downloads/>

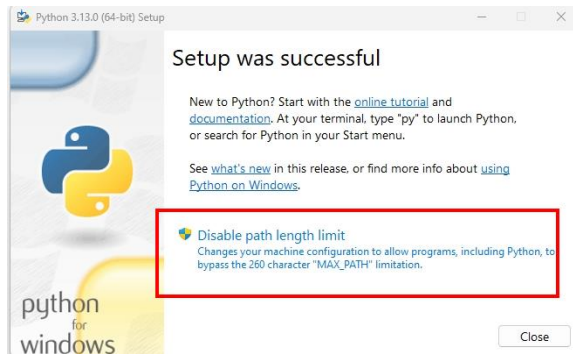


- เลือกเวอร์ชัน: เลือกเวอร์ชัน Python ที่เหมาะสมกับระบบปฏิบัติการของคุณ (Windows, macOS, หรือ Linux) โดยทั่วไปแนะนำให้เลือกเวอร์ชันล่าสุดที่รองรับ



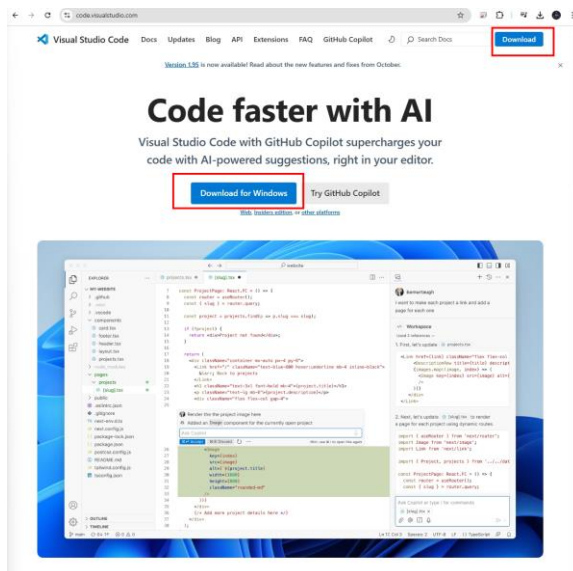
- ติดตั้ง: ทำตามขั้นตอนการติดตั้งตามปกติ **สำคัญ**: ตรวจสอบให้แน่ใจว่าได้เลือกตัวเลือก "Add Python 3.x to PATH" เพื่อให้ระบบสามารถค้นหา Python ได้จากทุกตำแหน่ง



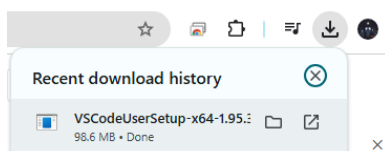


2. ดาวน์โหลดและติดตั้ง Visual Studio Code:

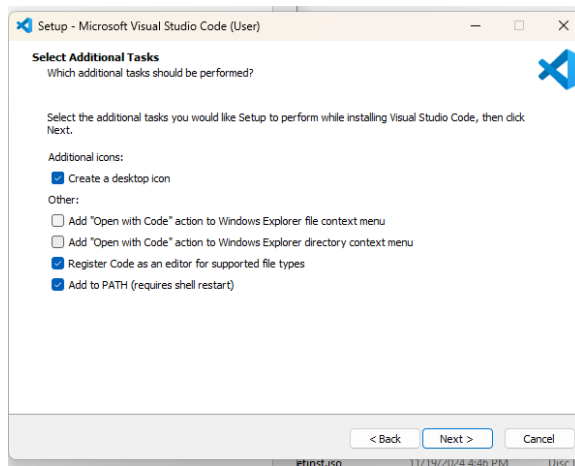
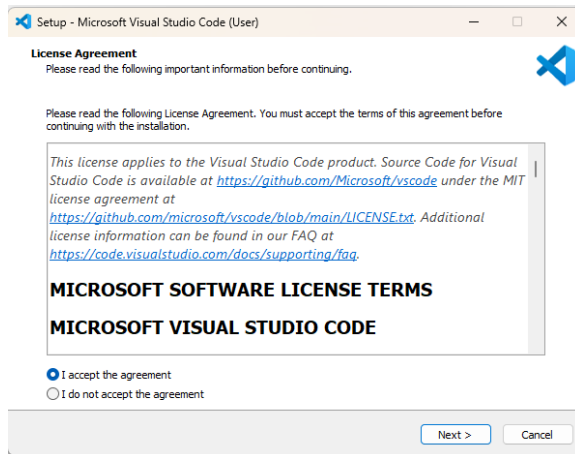
- เข้าเว็บไซต์ VS Code: ไปที่ <https://code.visualstudio.com/>

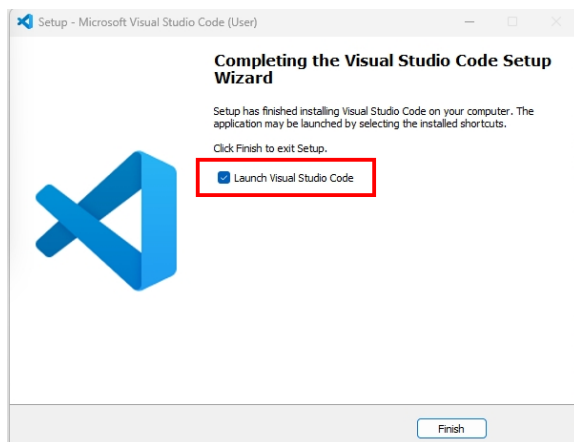
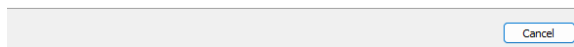
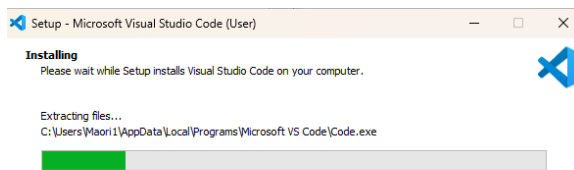
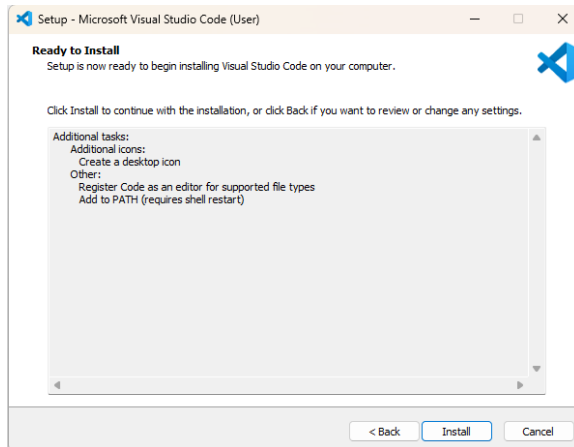


- เลือกเวอร์ชัน: เลือกเวอร์ชันที่ตรงกับระบบปฏิบัติการของคุณ
- ติดตั้ง: ทำตามขั้นตอนการติดตั้งตามปกติ



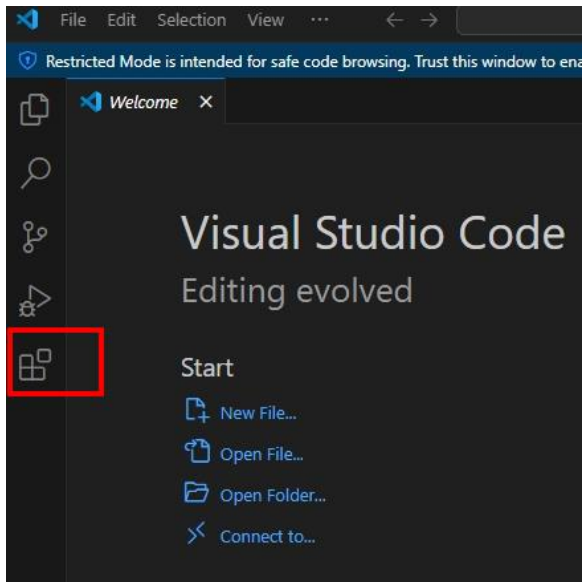
Name	Date modified	Type	Size
▼ Today			
VSCodeUserSetup-x64-1.95.3.exe	11/22/2024 9:09 PM	Application	100,978 KB
python-3.13.0-amd64 (1).exe	11/22/2024 9:02 PM	Application	27,501 KB
python-3.13.0-amd64.exe	11/22/2024 9:00 PM	Application	27,501 KB





3. ติดตั้ง Extension สำหรับ Python ใน VS Code:

- **เปิด VS Code:** หลังจากติดตั้งเสร็จสิ้น ให้เปิดโปรแกรมขึ้นมา
- **ค้นหา Extension:** คลิกที่ไอคอน Extensions บนแถบด้านข้างซ้าย (รูปสี่เหลี่ยมที่มีสี่เหลี่ยมเล็กๆ ภายใน)

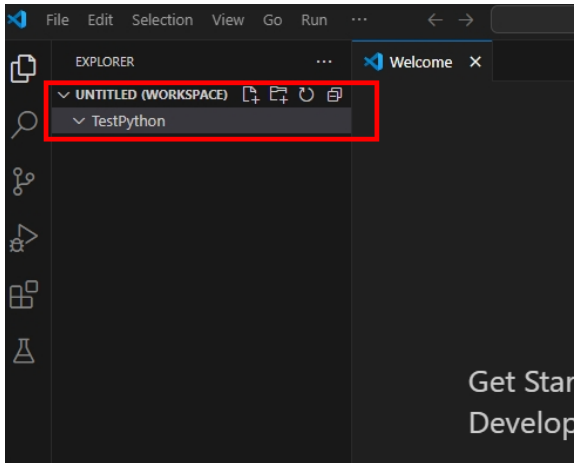


- ค้นหา **Python**: พิมพ์ "Python" ในช่องค้นหา
- ติดตั้ง **Extension**: คลิกปุ่ม Install บน Extension ที่มีชื่อว่า "Python" โดย Microsoft

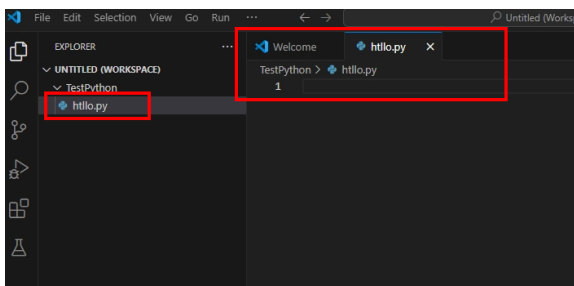


4. ตรวจสอบการติดตั้ง:

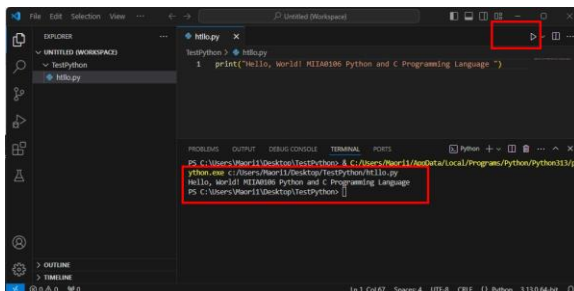
- สร้าง Folder TestPython
- เปิด Folder TestPython



- **สร้างไฟล์ Python:** สร้างไฟล์ใหม่ (File -> New File) แล้วบันทึกเป็นนามสกุล .py (เช่น hello.py)



- **เขียนโค้ด:** เขียนโค้ด Python ง่ายๆ เช่น



- **รันโค้ด:** คลิกขวาที่ไฟล์ แล้วเลือก "Run Python File in Terminal" หรือใช้ปุ่มลัด (มักจะเป็น Ctrl+Alt+N บน Windows และ Linux)

หากทุกอย่างถูกต้อง โปรแกรมจะแสดงผลลัพธ์ "Hello, World!" ใน Terminal

การติดตั้ง Visual Studio 2022 สำหรับการพัฒนา C++:

Visual Studio 2022 เป็น Integrated Development Environment (IDE) ที่ทรงพลังและได้รับความนิยมอย่างสูงสำหรับการพัฒนาซอฟต์แวร์หลากหลายภาษา รวมถึง C++ ด้วย ขั้นตอนการติดตั้ง Visual Studio 2022 เพื่อรองรับการพัฒนา C++ นั้นค่อนข้างง่าย เพียงแค่ทำตามขั้นตอนต่อไปนี้:

ขั้นตอนการติดตั้ง

1. ดาวน์โหลดตัวติดตั้ง:

- เข้าไปที่เว็บไซต์หลักของ Visual Studio: <https://visualstudio.microsoft.com/>
- คลิกที่ปุ่ม "ดาวน์โหลดฟรี" (Free download)
- เลือก **Community edition** (สำหรับการใช้งานส่วนบุคคลและโครงการโอเพนซอร์ส) หรือ **Professional/Enterprise edition** (สำหรับองค์กร) ตามความเหมาะสม

2. เรียกใช้ตัวติดตั้ง:

- คลิกสองครั้งที่ไฟล์ตัวติดตั้งที่ดาวน์โหลดมา
- รอให้ตัวติดตั้งเริ่มทำงาน

3. เลือก Workload:

- ในหน้าต่างตัวติดตั้ง เลือก **Desktop development with C++** เพื่อติดตั้งเครื่องมือที่จำเป็นสำหรับการพัฒนา C++
- หากต้องการติดตั้งส่วนประกอบอื่นๆ เพิ่มเติม เช่น .NET, Azure, หรือ Mobile development สามารถเลือกได้ในขั้นตอนนี้

4. ปรับแต่งการติดตั้ง:

- คุณสามารถปรับแต่งการติดตั้งได้เพิ่มเติม เช่น เลือกตำแหน่งในการติดตั้ง, เลือกส่วนประกอบที่ต้องการติดตั้ง หรือยกเลิกการติดตั้งส่วนประกอบที่ไม่ต้องการ

5. เริ่มการติดตั้ง:

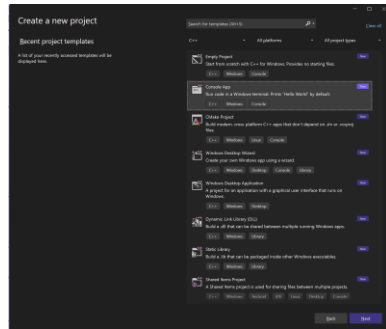
- คลิกปุ่ม **Install** เพื่อเริ่มกระบวนการติดตั้ง
- รอให้การติดตั้งเสร็จสิ้น

หลังจากติดตั้งเสร็จสิ้น

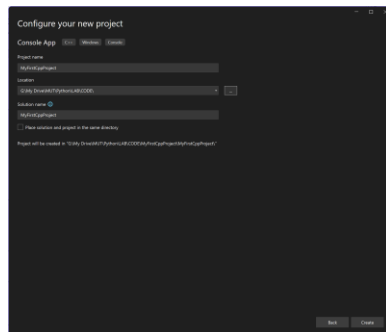
- **เปิด Visual Studio 2022:** ค้นหา Visual Studio 2022 จากเมนู Start แล้วเปิดโปรแกรม
- **สร้างโครงการ C++ ใหม่:** เลือกเมนู File -> New -> Project จากนั้นเลือกเทมเพลตโครงการ C++ ที่ต้องการ เช่น Console app, Windows Desktop app, หรือโครงการประเภทอื่นๆ
- **เริ่มเขียนโค้ด:** คุณสามารถเริ่มเขียนโค้ด C++ ได้ทันทีใน Visual Studio

1.1.2.สร้างโปรเจกต์ใหม่ใน C++

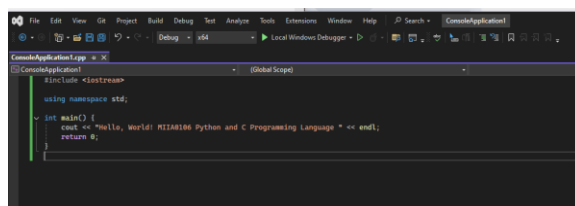
- 1) เปิด Visual Studio 2023.
- 2) คลิก "Create a new project".



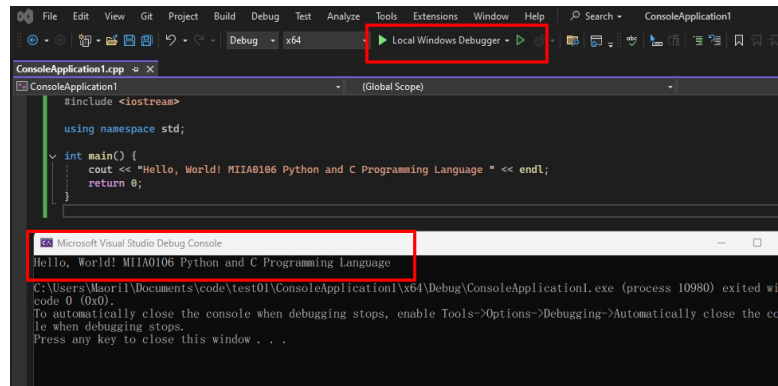
- 3) เลือกเทมเพลต "Console App" (C++) แล้วคลิก Next.
- 4) ตั้งชื่อโปรเจกต์ (เช่น MyFirstCppProject) และเลือกโฟลเดอร์สำหรับเก็บไฟล์โปรเจกต์.



- 5) คลิก Create เพื่อสร้างโปรเจกต์.
- 6) เขียน code เพื่อทดสอบ



- 7) รันโปรแกรม



1.1.3.สรุปคำสั่ง

เปรียบเทียบ C++ กับ Python		
คุณสมบัติ	C++	Python
รับข้อมูล (Input)	cin >>	input()
แสดงผล (Output)	cout <<	print()
การจัดฟอร์แมตข้อความ	ต้องใช้ << และระวังช่องว่าง	ใช้ f-string ง่ายและยืดหยุ่น
การแปลงข้อมูล (Casting)	ทำงานโดยตรง (int, float, ฯลฯ)	ต้องใช้ int(), float() ฯลฯ

<pre>if (เงื่อนไขที่ 1) { // ทำงานเมื่อเงื่อนไขที่ 1 เป็นจริง } else if (เงื่อนไขที่ 2) { // ทำงานเมื่อเงื่อนไขที่ 1 เป็นเท็จ แต่เงื่อนไขที่ 2 เป็นจริง } else if (เงื่อนไขที่ 3) { // ทำงานเมื่อเงื่อนไขที่ 1 และ 2 เป็นเท็จ แต่เงื่อนไขที่ 3 เป็นจริง } else { // ทำงานเมื่อทุกเงื่อนไขเป็นเท็จ }</pre>	<pre>if เงื่อนไข1: # คำสั่งเมื่อเงื่อนไขเป็นจริง elif เงื่อนไข2: # คำสั่งเมื่อเงื่อนไขที่ 1 เป็นเท็จ แต่เงื่อนไขที่ 2 เป็นจริง elif เงื่อนไข3: # คำสั่งเมื่อเงื่อนไขที่ 1 และ 2 เป็นเท็จ แต่เงื่อนไขที่ 3 เป็นจริง else: # คำสั่งเมื่อเงื่อนไขเป็นเท็จ</pre>
---	--

<pre>switch (ตัวแปร) { case ค่า1: // คำสั่งเมื่อค่าเท่ากับ ค่า1 break; case ค่า2: // คำสั่งเมื่อค่าเท่ากับ ค่า2 break; default: // คำสั่งเมื่อไม่ตรงกับกรณีใดเลย }</pre>	<pre>match ตัวแปร: case ค่า1: # คำสั่งที่ต้องการ case ค่า2: # คำสั่งที่ต้องการ case _: # คำสั่งเมื่อไม่ตรงกับกรณีใด ๆ</pre>
--	---

1.1.4.สร้างโปรแกรมที่มีเมนูให้ผู้ใช้เลือก

ตัวอย่าง

สร้างโปรแกรมที่มีเมนูให้ผู้ใช้เลือก 3 ตัวเลือก ได้แก่

- 1) หาพื้นที่ของสี่เหลี่ยมผืนผ้า
- 2) หาปริมาตรของสี่เหลี่ยมผืนผ้า
- 3) คำนวณเงินเดือนของพนักงาน
- 4) เมื่อผู้ใช้เลือกเมนู โปรแกรมจะแสดงชื่อเมนูที่เลือกก่อนเข้าสู่การทำงานของเมนูนั้น

1) วัตถุประสงค์ของการเขียนโปรแกรม

เพื่อพัฒนาโปรแกรมที่แสดงเมนูสำหรับเลือกฟังก์ชันต่าง ๆ ได้แก่:

- คำนวณพื้นที่ของสี่เหลี่ยมผืนผ้า
- คำนวณปริมาตรของสี่เหลี่ยมผืนผ้า
- คำนวณเงินเดือนของพนักงาน

โดยผู้ใช้สามารถเลือกเมนู และโปรแกรมจะแสดงชื่อเมนูที่เลือกก่อนดำเนินการ

2) รูปแบบผลลัพธ์ที่ต้องการ

1.เมื่อเริ่มโปรแกรม:

แสดงข้อความเมนูให้เลือก

โปรดเลือกเมนูที่ต้องการ:

1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า
2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า
3. โปรแกรมคำนวณเงินเดือนของพนักงาน

เลือก:

2.เมื่อผู้ใช้เลือกเมนู (เช่น เลือก 1):

คุณเลือก: โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า

3.กรณีป้อนตัวเลือกไม่ถูกต้อง:

ตัวเลือกไม่ถูกต้อง

3) ข้อมูลนำเข้า

เลขเมนูที่ผู้ใช้เลือก (1, 2, 3 หรือค่าอื่น)

ข้อมูลเพิ่มเติม (ขึ้นอยู่กับเมนูที่เลือก เช่น):

เมนู 1: ความกว้างและความยาวของสี่เหลี่ยมผืนผ้า

เมนู 2: ความกว้าง ความยาว และความสูงของสี่เหลี่ยมผืนผ้า

เมนู 3: อัตราเงินเดือนต่อเดือน และจำนวนชั่วโมงที่ทำงาน

4) ตัวแปรที่ใช้

C++:

int choice: เก็บตัวเลือกเมนูที่ผู้ใช้เลือก

Python:

choice: เก็บตัวเลือกเมนูที่ผู้ใช้เลือก

5) วิธีการประมวลผล

- 1) แสดงเมนู: ให้ผู้ใช้เลือกจาก 3 ตัวเลือก
- 2) รับค่า (Input): อ่านค่าตัวเลือกจากผู้ใช้
- 3) ตรวจสอบตัวเลือก (Processing):
 - a) หากเลือกเมนูที่ถูกต้อง:
 - i) แสดงชื่อเมนูที่เลือก
 - ii) ดำเนินการเพิ่มเติมตามเมนู (เช่น คำนวณพื้นที่, ปริมาตร, หรือเงินเดือน)
 - b) หากเลือกไม่ถูกต้อง:
 - i) แสดงข้อความแจ้งเตือน
- 4) แสดงผลลัพธ์ (Output): แสดงผลการเลือกหรือข้อความแจ้งเตือนข้อผิดพลาด

6.เขียน Flowchart

สอนในคาบ LEC

7.เขียนโปรแกรม

C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int choice;
    cout << "โปรดเลือกเมนูที่ต้องการ:\n";
    cout << "1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า\n";
    cout << "2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า\n";
    cout << "3. โปรแกรมคำนวณเงินเดือนของพนักงาน\n";
    cout << "เลือก: ";
    cin >> choice;

    switch (choice) {
        case 1:
            cout << "คุณเลือก: โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า\n";
            break;
        case 2:
            cout << "คุณเลือก: โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า\n";
            break;
        case 3:
            cout << "คุณเลือก: โปรแกรมคำนวณเงินเดือนของพนักงาน\n";
            break;
        default:
            cout << "ตัวเลือกไม่ถูกต้อง\n";
            break;
    }
    return 0;
}
```

Python Code

```
def main() :
    print("โปรดเลือกเมนูที่ต้องการ:")
    print("1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า")
    print("2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า")
    print("3. โปรแกรมคำนวณเงินเดือนของพนักงาน")

    choice = int(input("เลือก: "))

    if choice == 1:
        print("คุณเลือก: โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า")
    elif choice == 2 :
        print("คุณเลือก: โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า")
    elif choice == 3 :
        print("คุณเลือก: โปรแกรมคำนวณเงินเดือนของพนักงาน")
```

```

else:
print("ตัวเลือกไม่ถูกต้อง")

if __name__ == "__main__" :
    main()

```

เฉลี่ย

```

#include <iostream>
using namespace std;

int main() {
    int choice;
    cout << "Please select a menu option:\n";
    cout << "1. Program to calculate the area of a rectangle\n";
    cout << "2. Program to calculate the volume of a rectangular prism\n";
    cout << "3. Program to calculate employee salary\n";
    cout << "Select: ";
    cin >> choice;

    switch (choice) {
        case 1:
            cout << "You selected: Program to calculate the area of a rectangle\n";
            break;
        case 2:
            cout << "You selected: Program to calculate the volume of a rectangular
prism\n";
            break;
        case 3:
            cout << "You selected: Program to calculate employee salary\n";
            break;
        default:
            cout << "Invalid option\n";
            break;
    }
    return 0;
}

```

```

def main():
    print("Please select a menu option:")
    print("1. Program to calculate the area of a rectangle")
    print("2. Program to calculate the volume of a rectangular prism")
    print("3. Program to calculate employee salary")

    choice = int(input("Select: "))

    if choice == 1:
        print("You selected: Program to calculate the area of a rectangle")

```

```

elif choice == 2:
    print("You selected: Program to calculate the volume of a rectangular prism")
elif choice == 3:
    print("You selected: Program to calculate employee salary")
else:
    print("Invalid option")

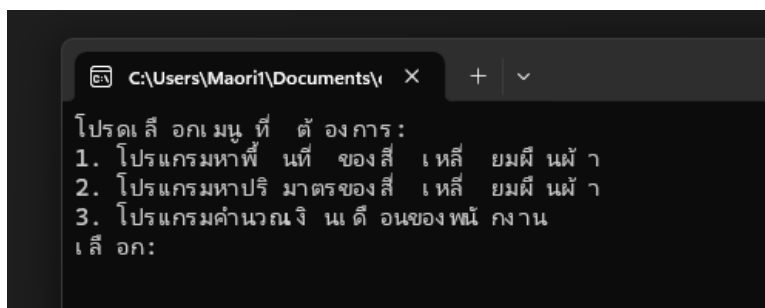
if __name__ == "__main__":
    main()

```

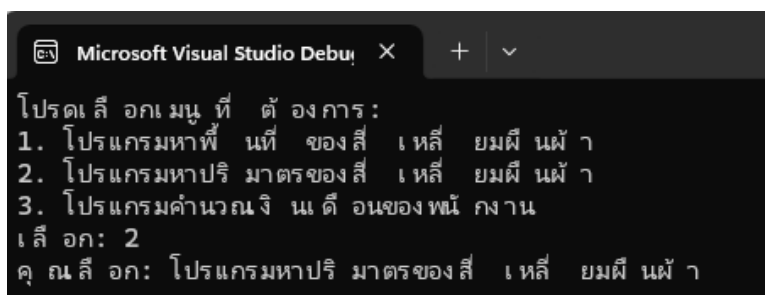
8.ผลลัพธ์

1.เมื่อเริ่มโปรแกรม:

แสดงข้อความเมนูให้เลือก



2.เมื่อผู้ใช้เลือกเมนู (เช่น เลือก 2):



3.กรณีป้อนตัวเลขไม่ถูกต้อง:

```
Microsoft Visual Studio Debug Console
โปรดเลือกเมนูที่ต้องการ :
1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า
2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า
3. โปรแกรมคำนวณเงินเดือนของพนักงาน
เลือก : 5
ตัวเลือกไม่ถูกต้อง
```

Python

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maori1\Desktop\TestPython> & C:/Users/Maori1/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Maori1/Desktop/TestPython/htllo.py
โปรดเลือกเมนูที่ต้องการ :
1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า
2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า
3. โปรแกรมคำนวณเงินเดือนของพนักงาน
เลือก : 1
คุณเลือก : โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า
PS C:\Users\Maori1\Desktop\TestPython> & C:/Users/Maori1/AppData/Local/Programs/Python/Python313/python.exe c:/Users/Maori1/Desktop/TestPython/htllo.py
โปรดเลือกเมนูที่ต้องการ :
1. โปรแกรมหาพื้นที่ของสี่เหลี่ยมผืนผ้า
2. โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า
3. โปรแกรมคำนวณเงินเดือนของพนักงาน
เลือก : 2
คุณเลือก : โปรแกรมหาปริมาตรของสี่เหลี่ยมผืนผ้า
PS C:\Users\Maori1\Desktop\TestPython>
```

1.1.5.เขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า

ให้นักศึกษาเขียนวิเคราะห์งานของการเขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า

1) วัตถุประสงค์ของการเขียนโปรแกรม

เพื่อพัฒนาโปรแกรมที่สามารถคำนวณ พื้นที่ของสี่เหลี่ยมผืนผ้า ได้โดยใช้ความกว้าง (Width) และความยาว (Length) ที่ผู้ใช้ป้อนเข้ามา พร้อมทั้งแสดงผลลัพธ์พื้นที่ (Area)

2) รูปแบบผลลัพธ์ที่ต้องการ

เมื่อผู้ใช้ป้อนค่าความกว้างและความยาว โปรแกรมจะแสดงผลลัพธ์พื้นที่ของสี่เหลี่ยมผืนผ้า

ตัวอย่างผลลัพธ์:

กรุณาใส่ค่าความกว้าง: 5 กรุณาใส่ค่าความยาว: 10 พื้นที่ของสี่เหลี่ยมผืนผ้าคือ: 50
--

3) ข้อมูลนำเข้า

1) ค่าที่ต้องการจากผู้ใช้:

- a) ความกว้าง (Width): ตัวเลข (float หรือ int)
- b) ความยาว (Length): ตัวเลข (float หรือ int)

4) ตัวแปรที่ใช้

1) C++:

- a) float width: เก็บค่าความกว้าง
- b) float length: เก็บค่าความยาว
- c) float area: เก็บค่าพื้นที่ที่คำนวณได้

2) Python:

- a) width: เก็บค่าความกว้าง
- b) length: เก็บค่าความยาว

c) area: เก็บค่าพื้นที่ที่คำนวณได้

5) วิธีการประมวลผล

1) รับค่าความกว้างและความยาวจากผู้ใช้:

a) ให้ผู้ใช้ป้อนค่าความกว้างและความยาวของสี่เหลี่ยมผืนผ้า

2) คำนวณพื้นที่ (Area):

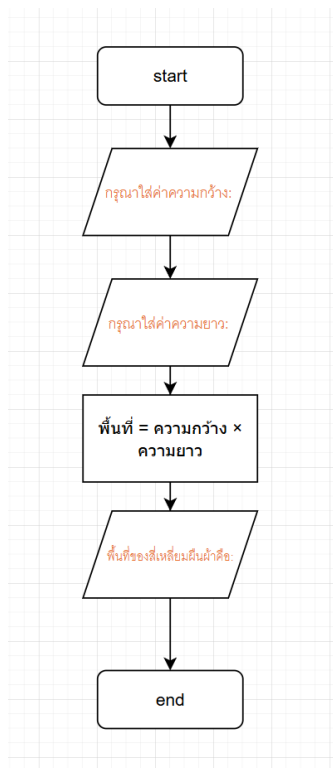
a) ใช้สูตร:

b) $\text{พื้นที่} = \text{ความกว้าง} \times \text{ความยาว}$

3) แสดงผลลัพธ์พื้นที่:

a) แสดงผลพื้นที่ในรูปแบบที่เข้าใจง่าย

6) ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งานของการเขียนโปรแกรมคำนวณพื้นที่สี่เหลี่ยมผืนผ้า



7.เขียนโปรแกรม

C++ Code

```
#include <iostream>
using namespace std;

int main() {
    float width, length, area;

    // รับค่าความกว้างและความยาว
    cout << "กรุณาใส่ค่าความกว้าง: ";
    cin >> _____;

    cout << "_____ : ";
    cin >> _____;

    // คำนวณพื้นที่
    area = width * length;

    // แสดงผลลัพธ์
    cout << "พื้นที่ของสี่เหลี่ยมผืนผ้าคือ: " << _____ << endl;

    return 0;
}
```

Python Code

```
# รับค่าความกว้าง ความยาว และความสูงจากผู้ใช้
width = float(input("กรุณาใส่ค่าความกว้าง: "))
length = float(input("_____ : "))
_____ = float(input("_____ : "))

# คำนวณปริมาตร
volume = _____

# แสดงผลลัพธ์
print(f"ปริมาตรของสี่เหลี่ยมผืนผ้าคือ: {_____}")
```

```

#include <iostream>
using namespace std;

int main() {
    float width, length, area;

    // Input width and length
    cout << "Enter the width: ";
    cin >> width;

    cout << "Enter the length: ";
    cin >> length;

    // Calculate the area
    area = width * length;

    // Display the result
    cout << "The area of the rectangle is: " << area << endl;

    return 0;
}

```

```

# Input width and length
width = float(input("Enter the width: "))
length = float(input("Enter the length: "))

# Calculate the area
area = width * length

# Display the result
print(f"The area of the rectangle is: {area}")

```

8.ผลลัพธ์

1.1.6.เขียนโปรแกรมคำนวณปริมาตรที่สี่เหลี่ยมผืนผ้า

1) วัตถุประสงค์ของการเขียนโปรแกรม

เพื่อพัฒนาโปรแกรมที่สามารถคำนวณ ปริมาตรของสี่เหลี่ยมผืนผ้า โดยรับค่าความกว้าง (Width), ความยาว (Length), และความสูง (Height) ที่ผู้ใช้ป้อน และแสดงผลลัพธ์ปริมาตร (Volume)

2) รูปแบบผลลัพธ์ที่ต้องการ

เมื่อผู้ใช้ป้อนค่าความกว้าง ความยาว และความสูง โปรแกรมจะแสดงผลลัพธ์ปริมาตรของสี่เหลี่ยมผืนผ้า

ตัวอย่างผลลัพธ์:

กรุณาใส่ค่าความกว้าง: 5 กรุณาใส่ค่าความยาว: 10 กรุณาใส่ค่าความสูง: 8 ปริมาตรของสี่เหลี่ยมผืนผ้าคือ: 400
--

3) ข้อมูลนำเข้า

1) ค่าที่ต้องการจากผู้ใช้:

- a) ความกว้าง (Width): ตัวเลข (float หรือ int)
- b) ความยาว (Length): ตัวเลข (float หรือ int)
- c) ความสูง (Height): ตัวเลข (float หรือ int)

4) ตัวแปรที่ใช้

1) C++:

- a) float width: เก็บค่าความกว้าง
- b) float length: เก็บค่าความยาว
- c) float height: เก็บค่าความสูง
- d) float volume: เก็บค่าปริมาตรที่คำนวณได้

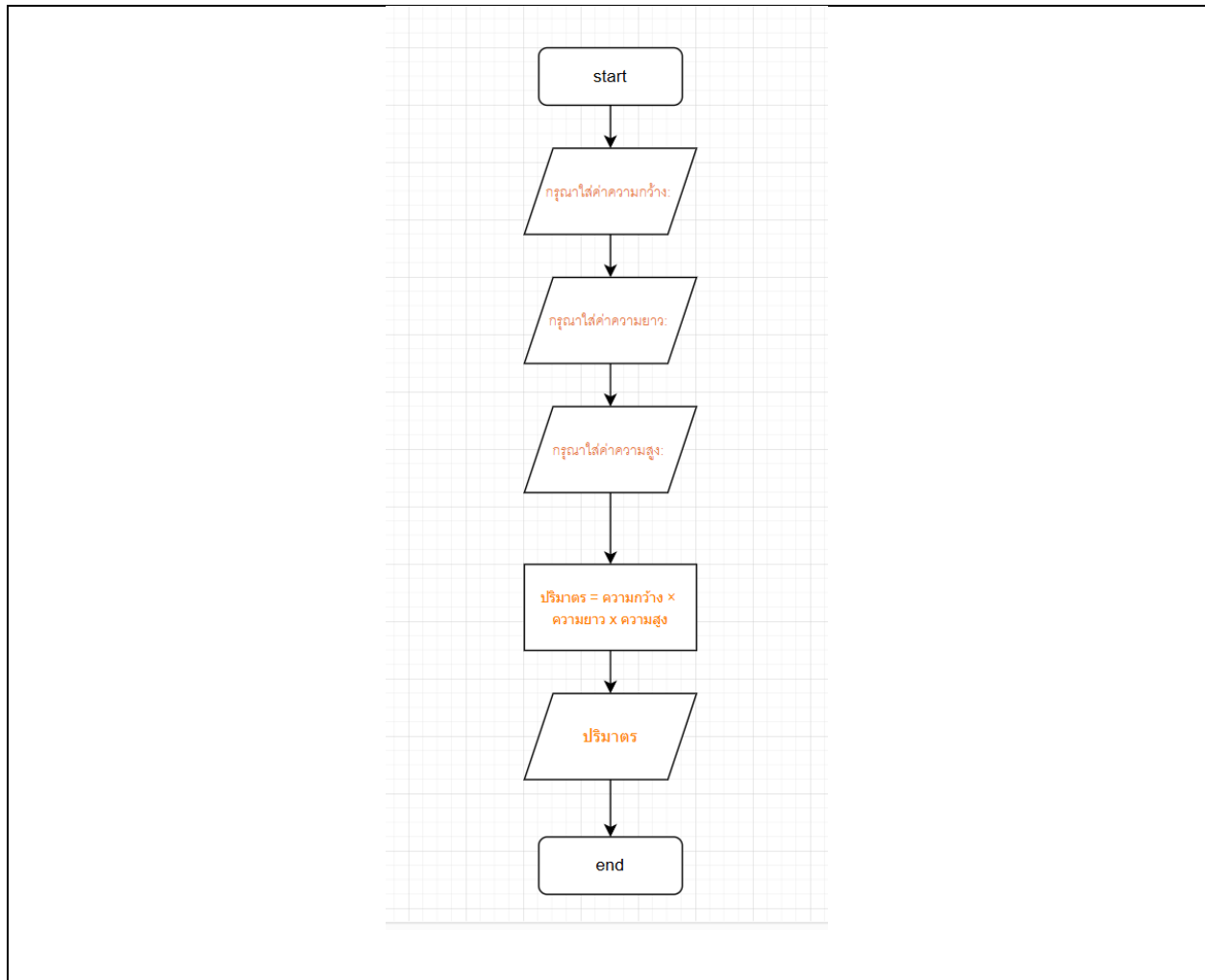
2) Python:

- a) width: เก็บค่าความกว้าง

- b) length: เก็บค่าความยาว
- c) height: เก็บค่าความสูง
- d) volume: เก็บค่าปริมาตรที่คำนวณได้

5) วิธีการประมวลผล

- 1) รับค่าความกว้าง ความยาว และความสูงจากผู้ใช้:
 - a) ให้ผู้ใช้ป้อนค่าความกว้าง, ความยาว, และความสูงของสี่เหลี่ยมผืนผ้า
- 2) คำนวณปริมาตร (Volume):
 - a) ใช้สูตร:
$$\text{ปริมาตร} = \text{ความกว้าง} \times \text{ความยาว} \times \text{ความสูง}$$
- 3) แสดงผลลัพธ์ปริมาตร:
 - a) แสดงผลปริมาตรในรูปแบบที่เข้าใจง่าย
- 6) ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน



7.เขียนโปรแกรม

C++ Code

```
#include <iostream>
using namespace std;

int main() {
    float width, length, height, volume;

    // Input width, length, and height
    cout << "Enter the width: ";
    cin >> width;

    cout << "Enter the length: ";
    cin >> length;

    cout << "Enter the height: ";
    cin >> height;

    // Calculate the volume
    volume = width * length * height;
```

```
// Display the result
cout << "The volume of the rectangular prism is: " << volume << endl;

return 0;
}
```

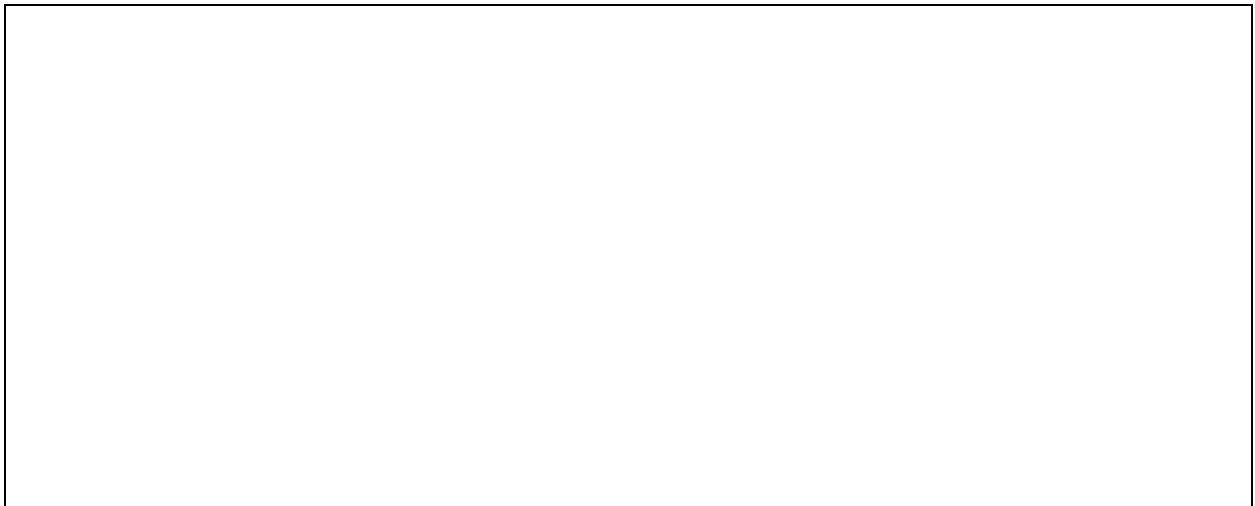
Python Code

```
# Input width, length, and height
width = float(input("Enter the width: "))
length = float(input("Enter the length: "))
height = float(input("Enter the height: "))

# Calculate the volume
volume = width * length * height

# Display the result
print(f"The volume of the rectangular prism is: {volume}")
```

8.ผลลัพธ์



1.1.7.การวิเคราะห์งานสำหรับโปรแกรมคำนวณเงินเดือนของพนักงาน

1) วัตถุประสงค์ของการเขียนโปรแกรม

เพื่อพัฒนาโปรแกรมสำหรับคำนวณเงินเดือนของพนักงาน โดยพิจารณาชั่วโมงการทำงานและอัตราค่าแรง พร้อมทั้งคำนวณภาษีและแสดงข้อมูลดังนี้:

- 1) รหัสพนักงาน
- 2) ค่าจ้างที่ยังไม่ถูกหักภาษี
- 3) จำนวนภาษีที่ต้องจ่าย
- 4) ค่าจ้างสุทธิที่พนักงานได้รับ

2) รูปแบบผลลัพธ์ที่ต้องการ

ตัวอย่างผลลัพธ์:

รหัสพนักงาน: E123 ค่าจ้างก่อนหักภาษี: 6000.00 บาท ภาษีที่ต้องจ่าย: 420.00 บาท ค่าจ้างสุทธิ: 5580.00 บาท
--

3) ข้อมูลนำเข้า

ค่าที่ต้องการจากผู้ใช้:

- 1) รหัสพนักงาน (string)
- 2) ชั่วโมงการทำงาน (float หรือ int)
- 3) อัตราค่าแรง (float หรือ int)

4) ตัวแปรที่ใช้

1) C++:

- a) string empID: เก็บรหัสพนักงาน
- b) float hoursWorked: เก็บชั่วโมงการทำงาน

- c) float hourlyRate: เก็บอัตราค่าแรง
- d) float grossSalary: ค่าจ้างก่อนหักภาษี
- e) float tax: ภาษีที่ต้องจ่าย
- f) float netSalary: ค่าจ้างสุทธิ

2) Python:

- a) emp_id: เก็บรหัสพนักงาน
- b) hours_worked: เก็บชั่วโมงการทำงาน
- c) hourly_rate: เก็บอัตราค่าแรง
- d) gross_salary: ค่าจ้างก่อนหักภาษี
- e) tax: ภาษีที่ต้องจ่าย
- f) net_salary: ค่าจ้างสุทธิ

5) วิธีการประมวลผล

1. รับค่าจากผู้ใช้: รหัสพนักงาน, ชั่วโมงการทำงาน, อัตราค่าแรง
2. คำนวณค่าจ้างก่อนหักภาษี (Gross Salary):
 - 2.1. ถ้าชั่วโมงการทำงาน < 30:

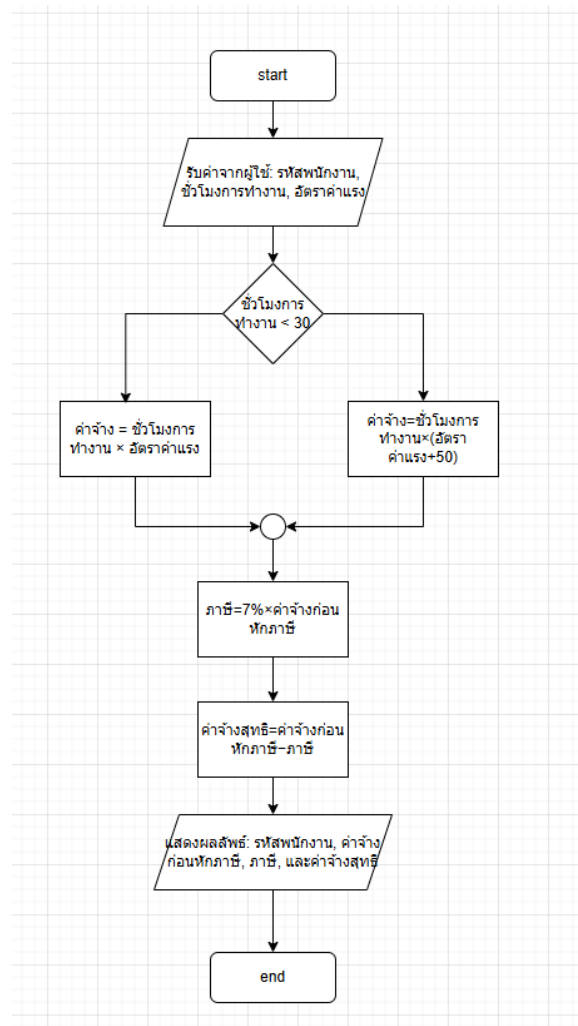
$$\text{ค่าจ้าง} = \text{ชั่วโมงการทำงาน} \times \text{อัตราค่าแรง}$$
 - 2.2. ถ้าชั่วโมงการทำงาน ≥ 30 :

$$\text{ค่าจ้าง} = \text{ชั่วโมงการทำงาน} \times (\text{อัตราค่าแรง} + 50)$$
3. คำนวณภาษี (Tax):

$$\text{ภาษี} = 7\% \times \text{ค่าจ้างก่อนหักภาษี}$$
4. คำนวณค่าจ้างสุทธิ (Net Salary):

$$\text{ค่าจ้างสุทธิ} = \text{ค่าจ้างก่อนหักภาษี} - \text{ภาษี}$$
5. แสดงผลลัพธ์: รหัสพนักงาน, ค่าจ้างก่อนหักภาษี, ภาษี, และค่าจ้างสุทธิ

6) ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการการวิเคราะห์งาน



7.เขียนโปรแกรม

C++ Code

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    string empID;
    float hoursWorked, hourlyRate, grossSalary, tax, netSalary;

    // Input data
    cout << "Enter employee ID: ";
    cin >> empID;

    cout << "Enter hours worked: ";
    cin >> hoursWorked;

    cout << "Enter hourly rate: ";
    cin >> hourlyRate;

    // Calculate gross salary
    if (hoursWorked < 30) {
        grossSalary = hoursWorked * hourlyRate;
    }
    else {
        grossSalary = hoursWorked * (hourlyRate + 50);
    }

    // Calculate tax and net salary
    tax = 0.07 * grossSalary;
    netSalary = grossSalary - tax;

    // Display the results
    cout << fixed << setprecision(2); // Set to display 2 decimal places
    cout << "Employee ID: " << empID << endl;
    cout << "Gross salary: " << grossSalary << " THB" << endl;
    cout << "Tax to be paid: " << tax << " THB" << endl;
    cout << "Net salary: " << netSalary << " THB" << endl;

    return 0;
}
```


Python Code

```
# Input data
emp_id = input("Enter employee ID: ")
hours_worked = float(input("Enter hours worked: "))
hourly_rate = float(input("Enter hourly rate: "))

# Calculate gross salary
if hours_worked < 30:
    gross_salary = hours_worked * hourly_rate
else:
    gross_salary = hours_worked * (hourly_rate + 50)

# Calculate tax and net salary
tax = 0.07 * gross_salary
net_salary = gross_salary - tax

# Display the results
print(f"Employee ID: {emp_id}")
print(f"Gross salary: {gross_salary:.2f} THB")
print(f"Tax to be paid: {tax:.2f} THB")
print(f"Net salary: {net_salary:.2f} THB")
```

8.ผลลัพธ์



1.1.8.การวิเคราะห์งานโปรแกรมเมนูคำนวณต่าง ๆ

เขียนโปรแกรมที่มีเมนูให้ผู้ใช้เลือก 3 ตัวเลือก ดังนี้:

- 1) คำนวณพื้นที่ของสี่เหลี่ยมผืนผ้า
- 2) คำนวณปริมาตรของสี่เหลี่ยมผืนผ้า
- 3) คำนวณเงินเดือนของพนักงาน

เมื่อผู้ใช้เลือกเมนู โปรแกรมจะทำการคำนวณและแสดงผลลัพธ์ตามเมนูที่เลือก

- 1) วัตถุประสงค์ของการเขียนโปรแกรม

- 1) เพื่อช่วยให้ผู้ใช้สามารถคำนวณข้อมูลที่เกี่ยวข้องกับพื้นที่, ปริมาตร, และการคำนวณเงินเดือนของพนักงานได้อย่างสะดวกและรวดเร็ว
- 2) รองรับการเลือกคำนวณจากเมนูเดียวกัน

- 2) รูปแบบผลลัพธ์ที่ต้องการ

เมื่อผู้ใช้เลือกเมนู ระบบจะแสดงผลลัพธ์การคำนวณของแต่ละเมนู

- 3) ข้อมูลนำเข้า

เมนูที่เลือก (1, 2, หรือ 3)
ข้อมูลเพิ่มเติมตามเมนู:
เมนู 1: ความกว้างและความยาว
เมนู 2: ความกว้าง, ความยาว, และความสูง
เมนู 3: รหัสพนักงาน, ชั่วโมงการทำงาน, และอัตราค่าแรง

- 5) ตัวแปรที่ใช้

C++:
int choice: เก็บตัวเลือกเมนู

ตัวแปรเพิ่มเติม:

float width, length, height: สำหรับคำนวณพื้นที่และปริมาตร

string empID: รหัสพนักงาน

float hoursWorked, hourlyRate, grossSalary, tax, netSalary: สำหรับเงินเดือน

Python:

choice: เก็บตัวเลือกเมนู

ตัวแปรเพิ่มเติม:

width, length, height: สำหรับคำนวณพื้นที่และปริมาตร

emp_id: รหัสพนักงาน

hours_worked, hourly_rate, gross_salary, tax, net_salary: สำหรับเงินเดือน

6) วิธีการประมวลผล

1) แสดงเมนูและรับค่าตัวเลือกจากผู้ใช้

2) ตรวจสอบเมนูที่เลือก:

a) เมนู 1: รับค่าความกว้างและความยาว แล้วคำนวณพื้นที่

i) $\text{พื้นที่} = \text{ความกว้าง} \times \text{ความยาว}$

b) เมนู 2: รับค่าความกว้าง, ความยาว, และความสูง แล้วคำนวณปริมาตร

i) $\text{ปริมาตร} = \text{ความกว้าง} \times \text{ความยาว} \times \text{ความสูง}$

c) เมนู 3: รับรหัสพนักงาน, ชั่วโมงการทำงาน, และอัตราค่าแรง คำนวณค่าจ้างตามชั่วโมงการทำงาน และหักภาษี

i) $\text{ค่าจ้าง} = \text{ชั่วโมงการทำงาน} \times \text{อัตราค่าแรง} (+50 \text{ สำหรับชั่วโมง} \geq 30)$

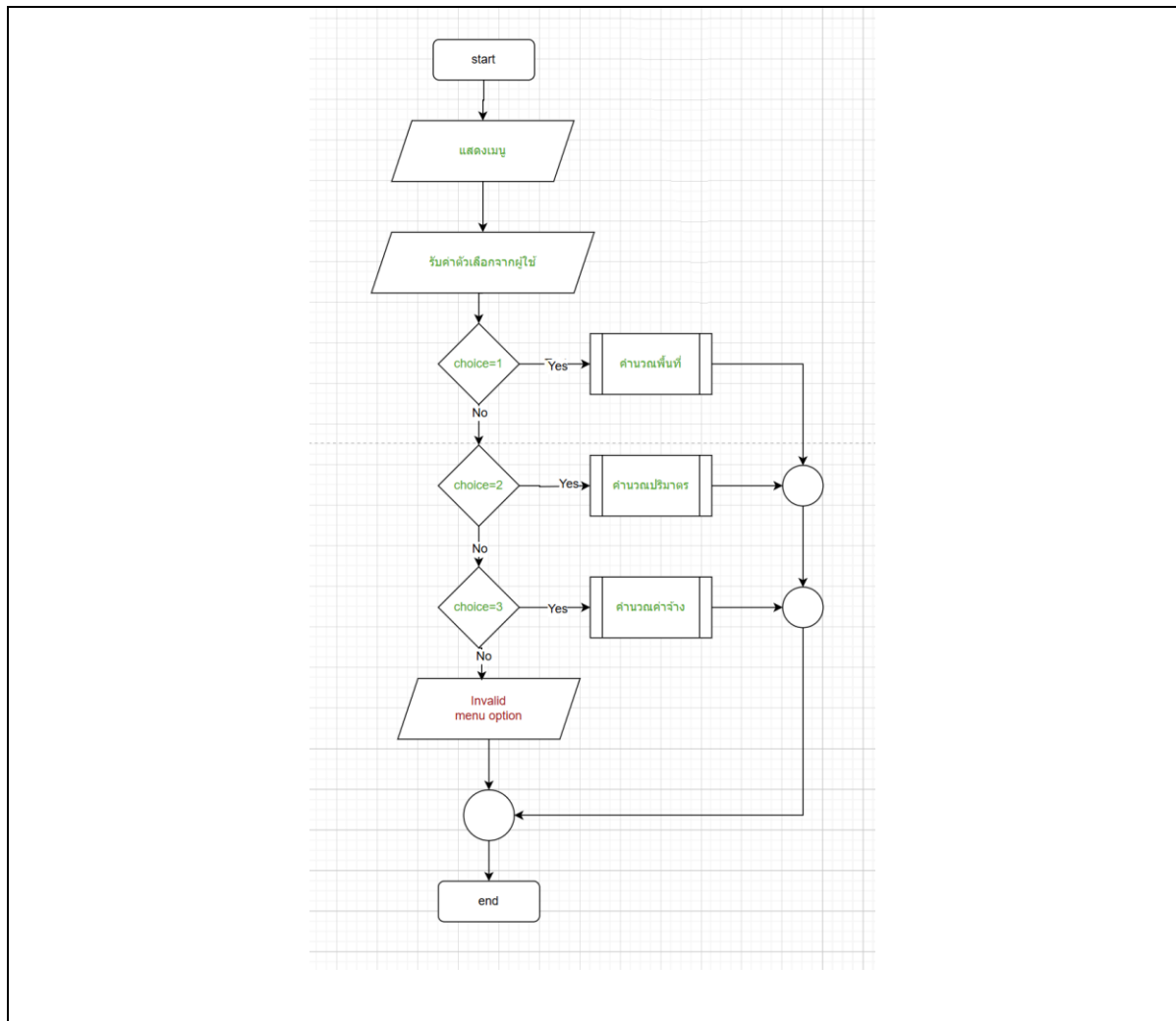
ii) $\text{ภาษี} = 7\% \times \text{ค่าจ้าง}$

iii) ค่าจ้างสุทธิ

iv) $\text{ค่าจ้างสุทธิ} = \text{ค่าจ้าง} - \text{ภาษี}$

แสดงผลลัพธ์ตามเมนูที่เลือก

7) ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการการวิเคราะห์งาน



8) เขียนโปรแกรม

C++ Code

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int choice;
    cout << "โปรดเลือกเมนู:\n";
    cout << "1. คำนวณพื้นที่ของสี่เหลี่ยมผืนผ้า\n";
    cout << "2. คำนวณปริมาตรของสี่เหลี่ยมผืนผ้า\n";
    cout << "3. คำนวณเงินเดือนของพนักงาน\n";
    cout << "เลือก: ";
    cin >> choice;

    if (choice == 1) {
        float width, length;
        cout << "กรุณาใส่ค่าความกว้าง: ";
        cin >> width;
        cout << "กรุณาใส่ค่าความยาว: ";
        cin >> length;
        cout << "พื้นที่ของสี่เหลี่ยมผืนผ้าคือ: " << width * length << endl;
    }
    else if (choice == 2) {
        _____
        _____
        _____
    }
    else if (choice == 3) {
        _____
        _____
        _____
    }
    else {
        cout << "เมนูไม่ถูกต้อง!" << endl;
    }

    return 0;
}
```

Python Code

```
def main():
    print("โปรดเลือกเมนู:")
    print("1. คำนวณพื้นที่ของสี่เหลี่ยมผืนผ้า")
    print("2. คำนวณปริมาตรของสี่เหลี่ยมผืนผ้า")
    print("3. คำนวณเงินเดือนของพนักงาน")
    choice = int(input("เลือก: "))

    if choice == 1:
        width = float(input("กรุณาใส่ค่าความกว้าง: "))
        length = float(input("กรุณาใส่ค่าความยาว: "))
        print(f"พื้นที่ของสี่เหลี่ยมผืนผ้าคือ: {width * length}")
    elif choice == 2:
        _____
        _____
        _____
    elif choice == 3:
        _____
        _____
        _____
    else:
        print("เมนูไม่ถูกต้อง!")

if __name__ == "__main__":
    main()
```

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    int choice;
    cout << "Please select a menu option:\n";
    cout << "1. Calculate the area of a rectangle\n";
    cout << "2. Calculate the volume of a rectangular prism\n";
    cout << "3. Calculate employee salary\n";
    cout << "Select: ";
    cin >> choice;

    if (choice == 1) {
        float width, length;
        cout << "Enter the width: ";
        cin >> width;
```

```

        cout << "Enter the length: ";
        cin >> length;
        cout << "The area of the rectangle is: " << width * length << endl;
    }
    else if (choice == 2) {
        float width, length, height;
        cout << "Enter the width: ";
        cin >> width;
        cout << "Enter the length: ";
        cin >> length;
        cout << "Enter the height: ";
        cin >> height;
        cout << "The volume of the rectangular prism is: " << width * length *
height << endl;
    }
    else if (choice == 3) {
        string empID;
        float hoursWorked, hourlyRate, grossSalary, tax, netSalary;

        cout << "Enter employee ID: ";
        cin >> empID;
        cout << "Enter hours worked: ";
        cin >> hoursWorked;
        cout << "Enter hourly rate: ";
        cin >> hourlyRate;

        if (hoursWorked < 30) {
            grossSalary = hoursWorked * hourlyRate;
        }
        else {
            grossSalary = hoursWorked * (hourlyRate + 50);
        }

        tax = 0.07 * grossSalary;
        netSalary = grossSalary - tax;

        cout << fixed << setprecision(2);
        cout << "Employee ID: " << empID << endl;
        cout << "Gross salary: " << grossSalary << " THB" << endl;
        cout << "Tax to be paid: " << tax << " THB" << endl;
        cout << "Net salary: " << netSalary << " THB" << endl;
    }
    else {
        cout << "Invalid menu option!" << endl;
    }

    return 0;
}

```

```

def main():
    print("Please select a menu option:")
    print("1. Calculate the area of a rectangle")
    print("2. Calculate the volume of a rectangular prism")

```

```

print("3. Calculate employee salary")

choice = int(input("Select: "))

if choice == 1:
    width = float(input("Enter the width: "))
    length = float(input("Enter the length: "))
    print(f"The area of the rectangle is: {width * length}")
elif choice == 2:
    width = float(input("Enter the width: "))
    length = float(input("Enter the length: "))
    height = float(input("Enter the height: "))
    print(f"The volume of the rectangular prism is: {width * length * height}")
elif choice == 3:
    emp_id = input("Enter employee ID: ")
    hours_worked = float(input("Enter hours worked: "))
    hourly_rate = float(input("Enter hourly rate: "))

    if hours_worked < 30:
        gross_salary = hours_worked * hourly_rate
    else:
        gross_salary = hours_worked * (hourly_rate + 50)

    tax = 0.07 * gross_salary
    net_salary = gross_salary - tax

    print(f"Employee ID: {emp_id}")
    print(f"Gross salary: {gross_salary:.2f} THB")
    print(f"Tax to be paid: {tax:.2f} THB")
    print(f"Net salary: {net_salary:.2f} THB")
else:
    print("Invalid menu option!")

if __name__ == "__main__":
    main()

```


9) ผลลัพธ์

A large, empty rectangular box with a thin black border, occupying the central portion of the page. It is intended for a drawing or a detailed written response.

1.1.9.เขียนโปรแกรมตรวจสอบเลขคู่เลขคี่

1.วัตถุประสงค์ของการเขียนโปรแกรม

- 1) เพื่อพัฒนาโปรแกรมที่สามารถตรวจสอบว่าจำนวนที่ผู้ใช้ป้อนเข้าไปเป็นเลขคู่ (Even) หรือเลขคี่ (Odd)
- 2) เพื่อช่วยในการเรียนรู้และทำความเข้าใจเกี่ยวกับการใช้ตัวดำเนินการทางคณิตศาสตร์ เช่น การหารเอาเศษ (%)

2.รูปแบบผลลัพธ์ที่ต้องการ

- 1) โปรแกรมรับตัวเลขจากผู้ใช้ และแสดงข้อความระบุว่าตัวเลขนั้นเป็นเลขคู่หรือเลขคี่
- 2) ตัวอย่างผลลัพธ์:
 - a) Input: Enter a number: 4
 - b) Output: 4 is an even number.
 - c) Input: Enter a number: 7
 - d) Output: 7 is an odd number.

3.ข้อมูลนำเข้า

จำนวนเต็ม (Integer) ที่ผู้ใช้ป้อน

4.ตัวแปรที่ใช้

C++:

int number: เก็บตัวเลขที่ผู้ใช้ป้อน

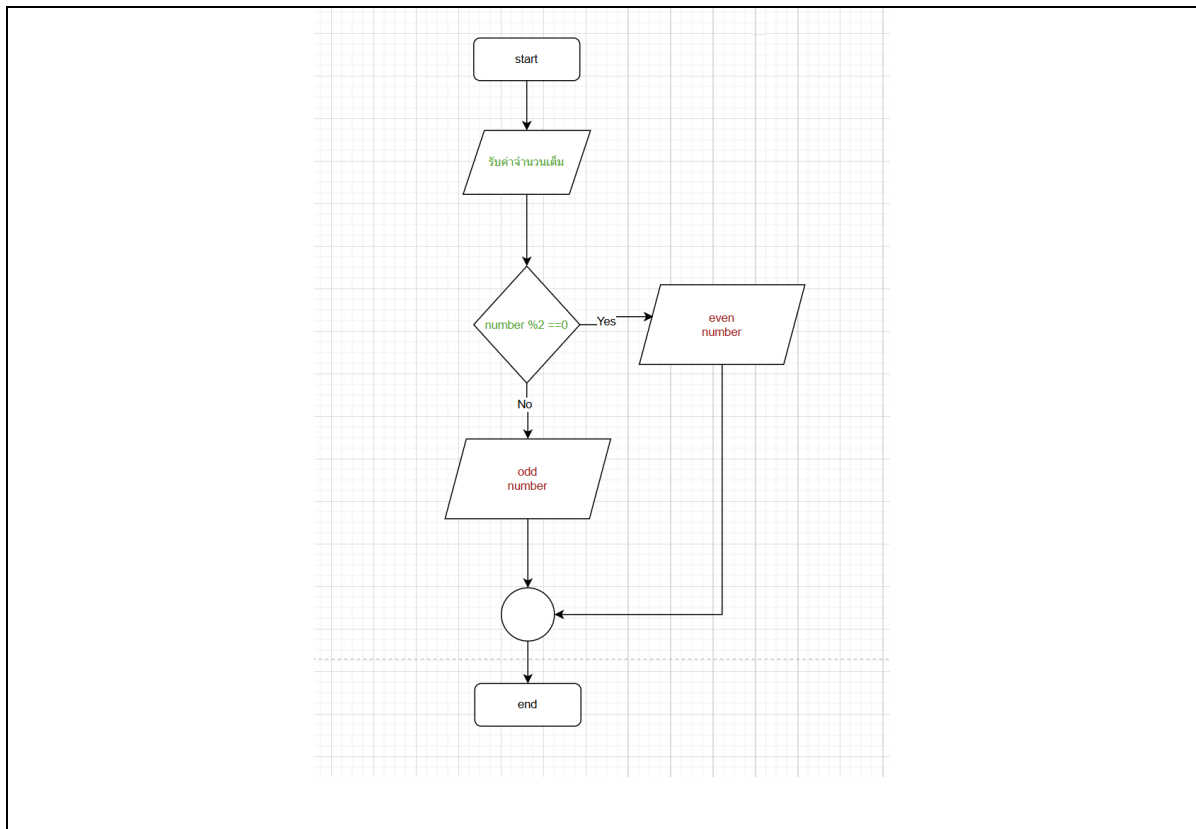
Python:

number: เก็บตัวเลขที่ผู้ใช้ป้อน

5.วิธีการประมวลผล

- 1) รับค่าจำนวนเต็มจากผู้ใช้
- 2) ตรวจสอบว่าจำนวนที่ป้อนเข้าไปเป็นเลขคู่หรือเลขคี่:
 - a) ใช้การหารเอาเศษ (%):
 - b) ถ้าค่า $\% 2 == 0 \rightarrow$ เป็นเลขคู่
 - c) ถ้าค่า $\% 2 \neq 0 \rightarrow$ เป็นเลขคี่
- 3) แสดงผลลัพธ์

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการการวิเคราะห์งาน



7)เขียนโปรแกรม

C++ Code

```
#include <iostream>
using namespace std;

int main() {
    int number;

    // Input: รับค่าจำนวนเต็มจากผู้ใช้
    cout << "Enter a number: ";
    cin >> number;

    // Processing: ตรวจสอบเลขคู่หรือเลขคี่
    if (number % 2 == 0) {
        cout << number << " is an even number." << endl;
    }
    else {
        cout << number << " is an odd number." << endl;
    }

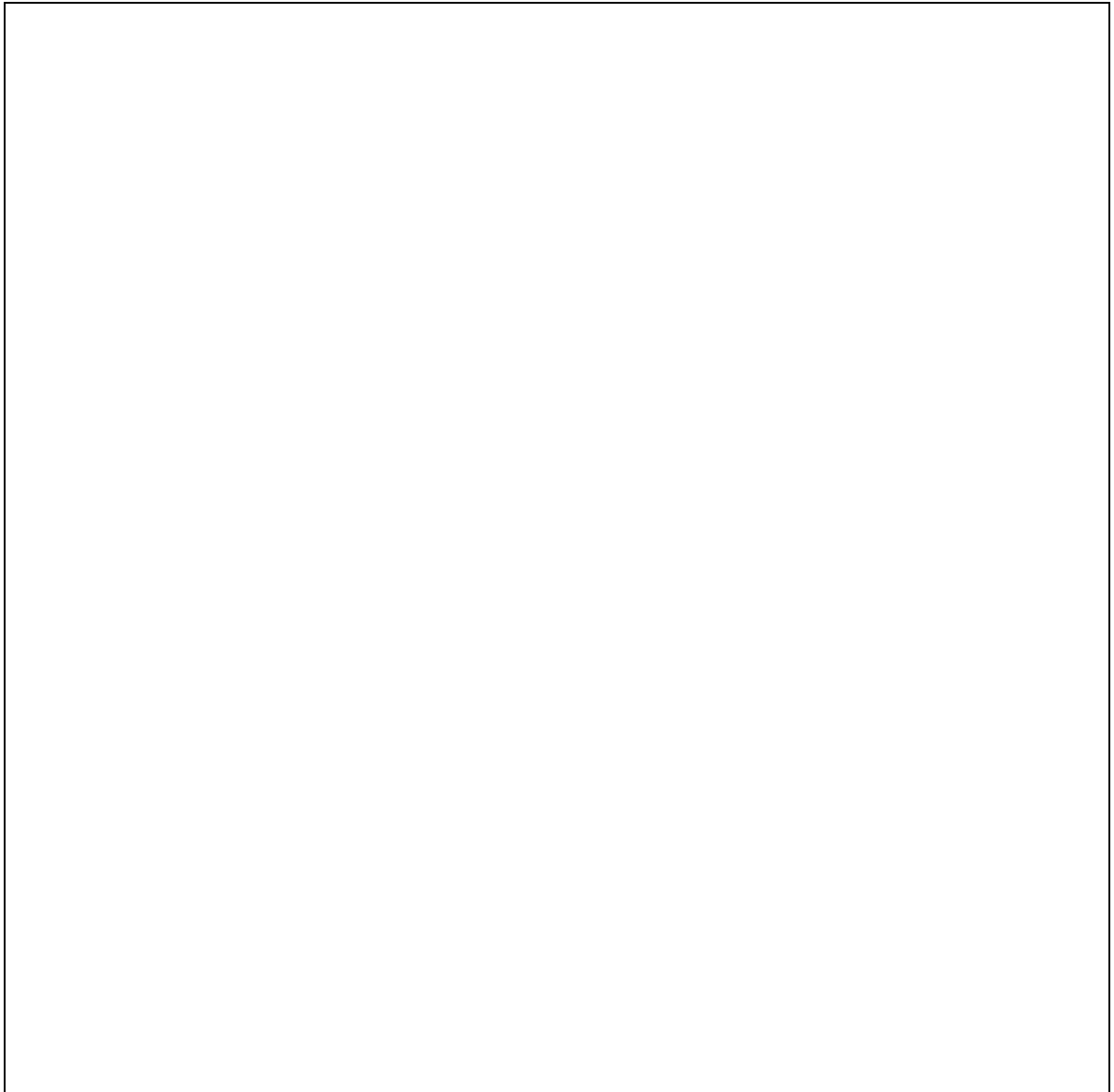
    return 0;
}
```

Python Code

```
# Input: รับค่าจำนวนเต็มจากผู้ใช้
number = int(input("Enter a number: "))

# Processing: ตรวจสอบเลขคู่หรือเลขคี่
if number % 2 == 0:
    print(f"{number} is an even number.")
else:
    print(f"{number} is an odd number.")
```

8)ผลลัพธ์

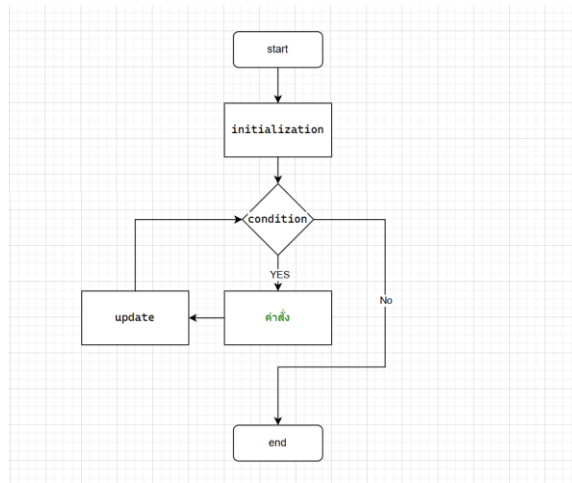


แบบฝึกหัดพื้นฐาน วนลูป (1-2 ชั่วโมง)

สรุปโครงสร้างคำสั่งของวงลูป (Loop)

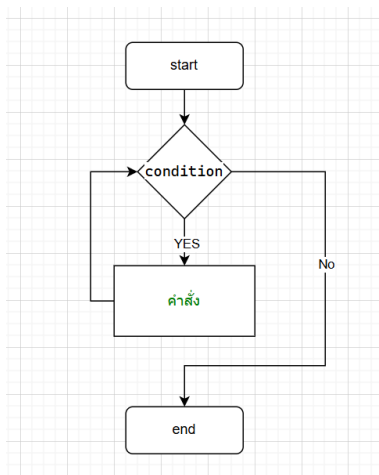
1. For Loop

โครงสร้างของ For Loop



<pre>for (initialization; condition; update) { // คำสั่งที่ต้องการทำซ้ำ }</pre>	<pre>for variable in iterable: # คำสั่งที่ต้องการทำซ้ำ</pre>
---	--

2 โครงสร้างของ While Loop

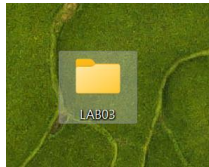


<pre>while (เงื่อนไข) { // โค้ดที่ต้องการทำซ้ำ }</pre>	<pre>while เงื่อนไข: # โค้ดที่ต้องการทำซ้ำ</pre>
--	--

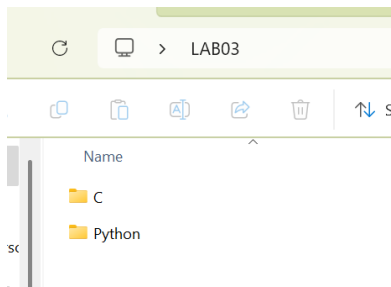
3. Do-While Loop (เฉพาะใน C++)

<pre>do { // คำสั่งที่ต้องการทำซ้ำ } while (condition);</pre>

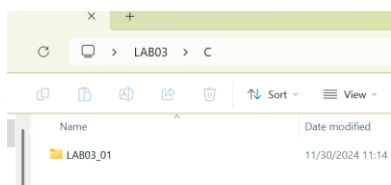
1.สร้าง Folder ชื่อว่า LAB03



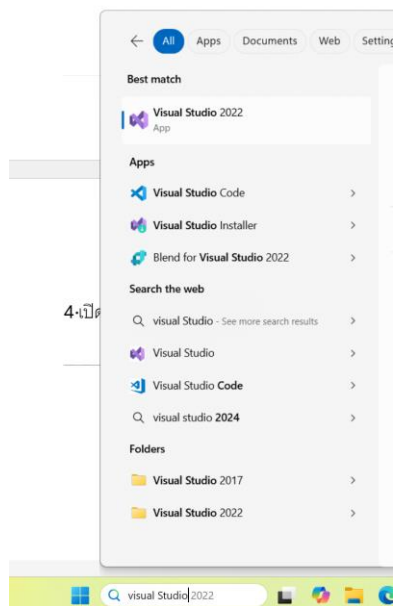
2.สร้าง Folder “C” และ “Python”



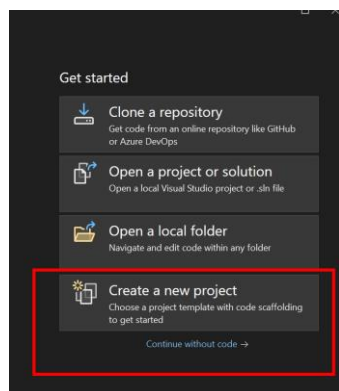
3.เข้าไปใน Folder C และสร้าง Folder “LAB03-01”



4 เปิดโปรแกรม visual Studio



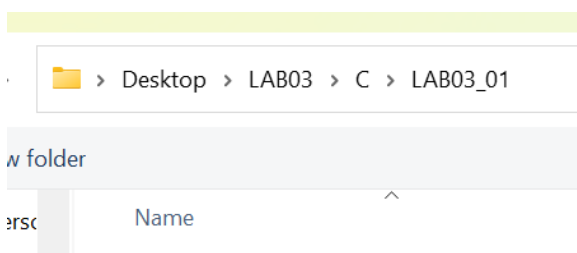
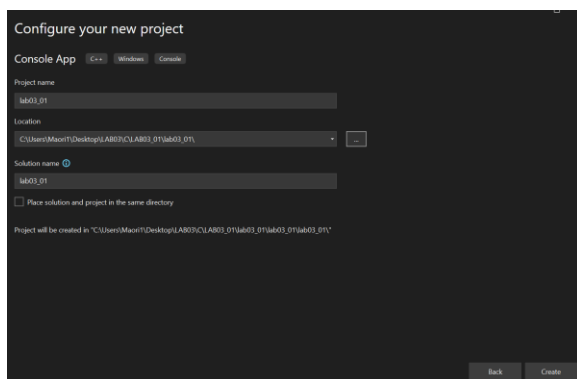
5 Create a new project



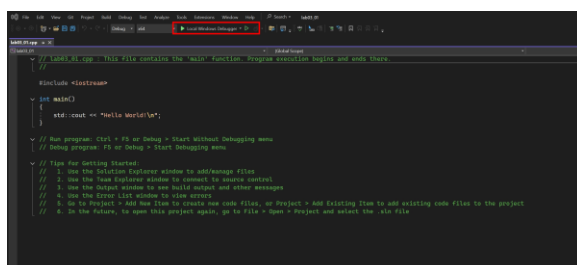
6.เลือก Console Project และ กด Next



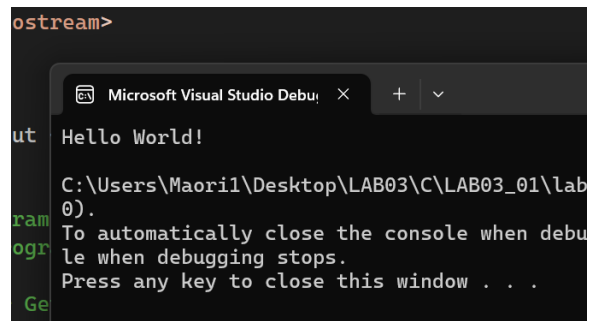
7. projectname lab03_01 และเลือกตำแหน่งของไฟล์ C:\Users\Maori1\Desktop\LAB03\C\LAB03_01 และกด Create



8. ทดสอบ รันโปรแกรม



9.ผลของการรันโปรแกรม



The screenshot shows a Visual Studio Debug Console window. The title bar reads "Microsoft Visual Studio Debug Console". The output text is as follows:

```
ostream>  
  
ut Hello World!  
C:\Users\Maori1\Desktop\LAB03\C\LAB03_01\lab  
0).  
To automatically close the console when debu  
le when debugging stops.  
Press any key to close this window . . .
```

1.1.10. พิมพ์เลข 1 ถึง 10 ด้วย for, while, และ do-while

1.วัตถุประสงค์ของการเขียนโปรแกรม

--

2.รูปแบบผลลัพธ์ที่ต้องการ

--

3.ข้อมูลนำเข้า

--

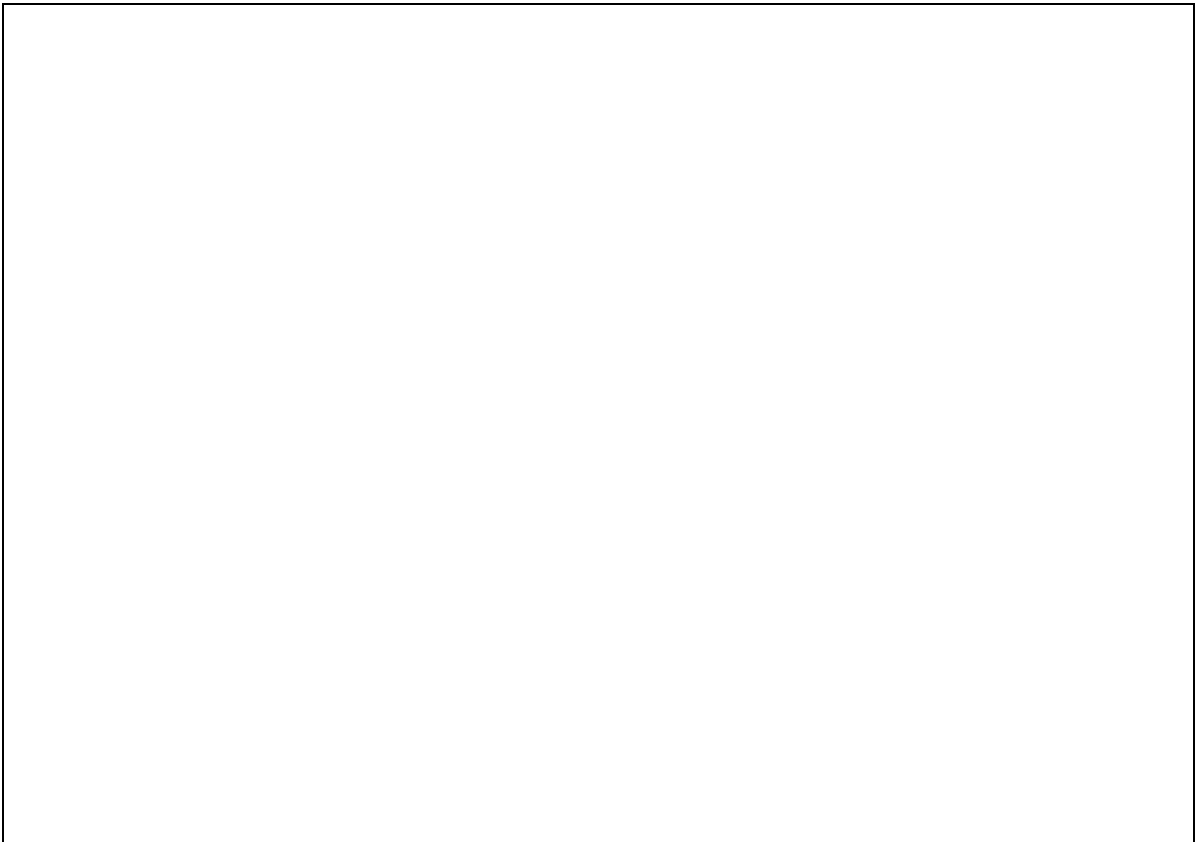
4.ตัวแปรที่ใช้

--

5.วิธีการประมวลผล



6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน



7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



1.1.11. พิมพ์เลขคู่ตั้งแต่ 2 ถึง 20

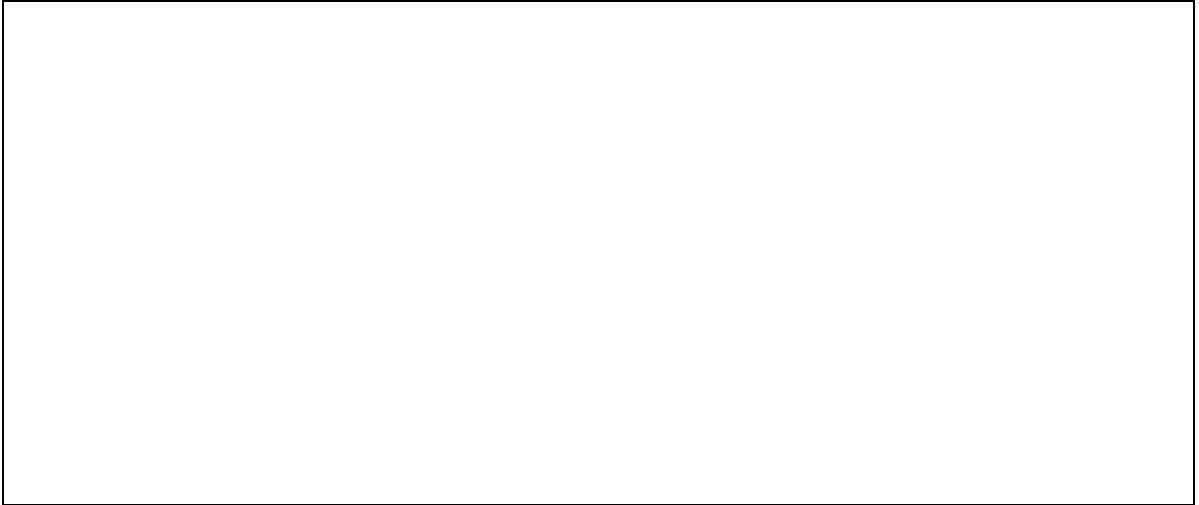
วัตถุประสงค์ของการเขียนโปรแกรม

2.รูปแบบผลลัพธ์ที่ต้องการ

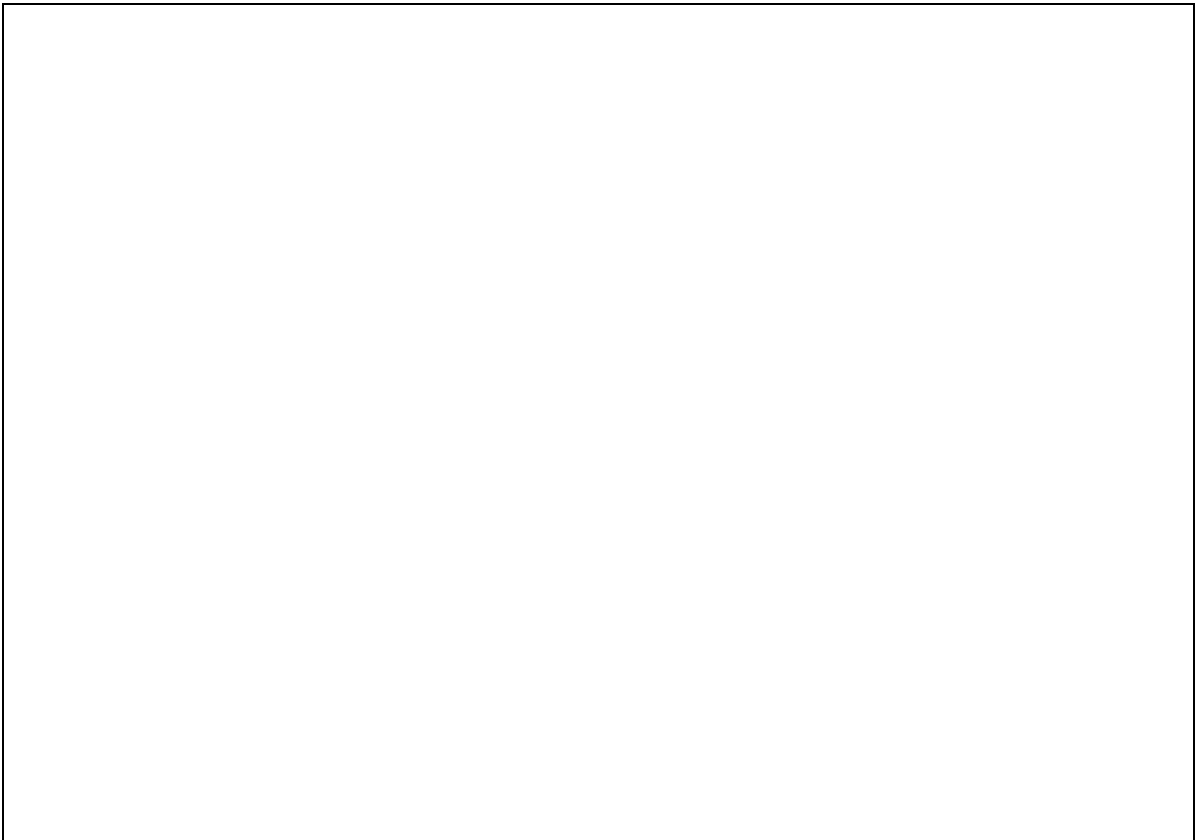
3.ข้อมูลนำเข้า

4.ตัวแปรที่ใช้

5.วิธีการประมวลผล



6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน

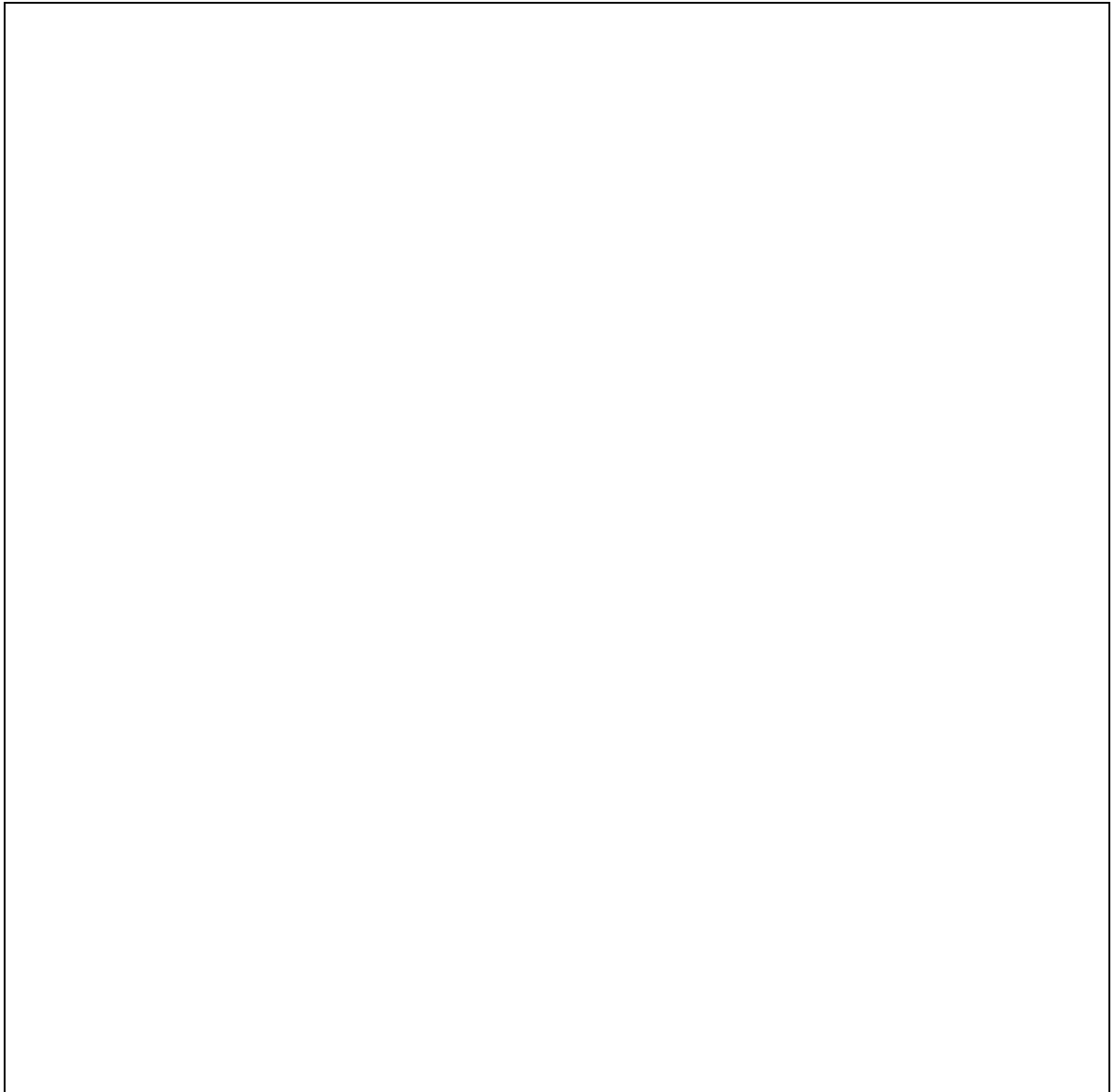


7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



1.1.12. หาผลรวมของเลข 1 ถึง 100

วัตถุประสงค์ของการเขียนโปรแกรม

--

2.รูปแบบผลลัพธ์ที่ต้องการ

--

3.ข้อมูลนำเข้า

--

4.ตัวแปรที่ใช้

--

5.วิธีการประมวลผล

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน

7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



1.1.13. เขียนโปรแกรมสร้างตารางสูตรคูณ ของเลขท้าย2หลักของรหัสนักศึกษา
วัตถุประสงค์ของการเขียนโปรแกรม

--

2.รูปแบบผลลัพธ์ที่ต้องการ

--

3.ข้อมูลนำเข้า

--

4.ตัวแปรที่ใช้

--

5.วิธีการประมวลผล

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน

7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



1.1.14. สร้างโปรแกรมเกมทายเลข (กลุ่มเลข 1-100 แล้วให้ผู้ใช้ทาย)

วัตถุประสงค์ของการเขียนโปรแกรม

--

2.รูปแบบผลลัพธ์ที่ต้องการ

--

3.ข้อมูลนำเข้า

--

4.ตัวแปรที่ใช้

--

5.วิธีการประมวลผล

--

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน



7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



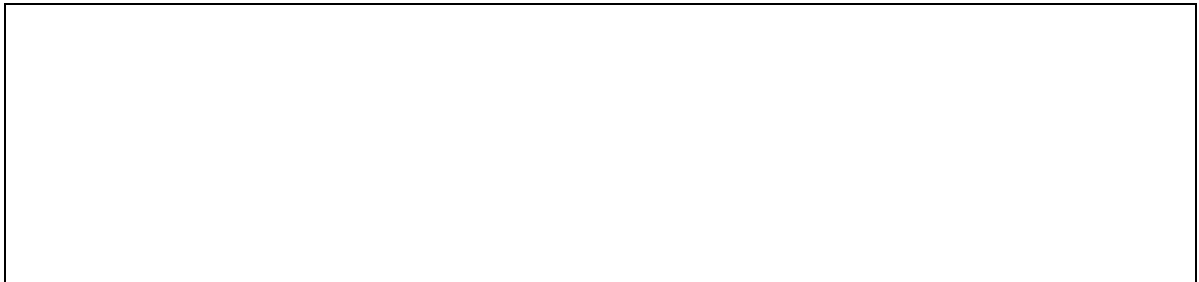
1.1.15. ระบบควบคุมรถวิ่งตามเส้น (Line Following Robot)

คำอธิบาย:

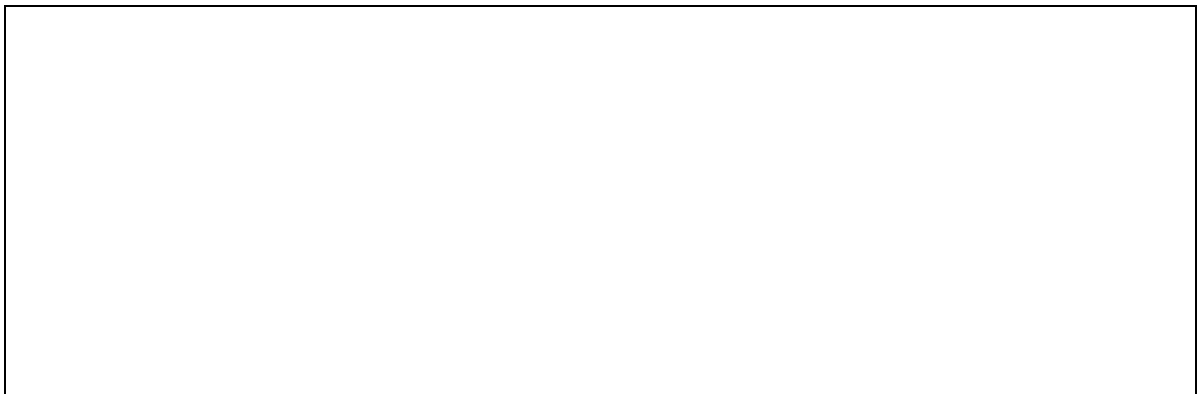
รถวิ่งตามเส้นต้องการวิ่งตามเส้นสีดำบนพื้นสีขาว รถมีเซนเซอร์สี (color sensor) ที่ติดตั้งที่ด้านหน้าของรถ เซนเซอร์จะตรวจจับสีและส่งข้อมูลกลับมาว่าเส้นสีดำอยู่ทางซ้าย, ขวา, หรือกลาง ระบบควบคุมจะตรวจสอบข้อมูลจากเซนเซอร์และทำการควบคุมรถให้เคลื่อนไหวตามเงื่อนไขต่อไปนี้:

1. ถ้าเส้นสีดำอยู่ทางซ้าย ให้รถเลี้ยวไปทางซ้าย
2. ถ้าเส้นสีดำอยู่ทางขวา ให้รถเลี้ยวไปทางขวา
3. ถ้าเส้นสีดำอยู่กลาง ให้รถวิ่งตรงไปข้างหน้า

วัตถุประสงค์ของการเขียนโปรแกรม



2.รูปแบบผลลัพธ์ที่ต้องการ



3.ข้อมูลนำเข้า

--

4.ตัวแปรที่ใช้

--

5.วิธีการประมวลผล

--

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน



7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์



1.1.16. สร้างฟังก์ชันเพื่อตรวจสอบว่าจำนวนใดเป็นจำนวนเฉพาะหรือไม่

วัตถุประสงค์ของการเขียนโปรแกรม

2.รูปแบบผลลัพธ์ที่ต้องการ

3.ข้อมูลนำเข้า

4.ตัวแปรที่ใช้

5.วิธีการประมวลผล

6.ให้นักศึกษาเขียนผังงานโปรแกรม ที่ได้จากการวิเคราะห์งาน

7)เขียนโปรแกรม

C++ Code

Python Code

8)ผลลัพธ์

